

فصل 5: کار کردن با توابع متنی

فصل 5: کار کردن با توابع متنی

در این فصل، شما:

- مروری بر مجموعه وسیع توابع مرتبط با متن در اکسل خواهید داشت.
- با کاراکترها و کدهای ANSI کار خواهید کرد.
- رشته‌ها را با تبدیل، فرمت‌بندی، تمیز کردن، پیوستن و استخراج متن دست‌کاری خواهید کرد.
- جستجوهای حساس به حالت و غیرحساس به حالت را برای زیررشته‌ها درون رشته‌های بزرگ‌تر انجام خواهید داد.
- یک زیررشته را با زیررشته دیگری جایگزین خواهید کرد.

در اکسل، متن به هر مجموعه‌ای از کاراکترهای الفبایی - عددانی گفته می‌شود که یک مقدار عددی، یک مقدار تاریخ یا زمان، یا یک فرمول نیست. کلمات، نام‌ها و برچسب‌ها همه به‌وضوح مقادیر متنی هستند، اما مقادیر سلول‌هایی که با آپاستروف (') شروع می‌شوند یا به‌عنوان متن فرمت‌بندی شده‌اند نیز متن هستند. مقادیر متنی همچنین به‌عنوان رشته‌ها شناخته می‌شوند و من از هر دو اصطلاح به‌طور متناوب در این فصل استفاده می‌کنم.

در فصل 1، "ساخت فرمول‌های بنیادی"، شما در مورد ساخت فرمول‌های متنی در اکسل یاد گرفتید—اینکه چیز زیادی برای یاد گرفتن وجود نداشت. فرمول‌های متنی فقط شامل عملگر الحاق (&) هستند که برای ترکیب دو یا چند رشته به یک رشته بزرگتر استفاده می‌شوند. توابع متنی اکسل به شما امکان می‌دهند تا فرمول‌های متنی را به سطحی مفیدتر برسانید و روش‌های متعددی برای دستکاری رشته‌ها در اختیارتان قرار می‌دهند. با این توابع، می‌توانید اعداد را به رشته‌ها تبدیل کنید، حروف کوچک را به حروف بزرگ (و بالعکس) تغییر دهید، دو رشته را مقایسه کنید و بیشتر.

توابع متنی اکسل

جدول 5-1 توابع متنی اکسل را خلاصه می‌کند و بقیه این فصل جزئیات و مثال‌هایی از نحوه استفاده از بیشتر آن‌ها را به شما می‌دهد.

فصل 5: کارکردن با توابع متنی

جدول 1-5 توابع متنی اکسل

Function	Description
ARRAYTOTEXT (<i>array</i> [, <i>format</i>])	Returns the text representation of <i>array</i> .
BAHTTEXT (<i>number</i>)	Converts <i>number</i> to baht (Thai) text.
CHAR (<i>number</i>)	Returns the character that corresponds to the ANSI code given by <i>number</i> .

CLEAN (<i>text</i>)	Removes all nonprintable characters from <i>text</i> .
CODE (<i>text</i>)	Returns the ANSI code for the first character in <i>text</i> .
CONCAT (<i>text1</i> [, <i>text2</i>] , ...)	Joins the specified strings into a single string.
DOLLAR (<i>number</i> [, <i>decimals</i>])	Converts <i>number</i> to a string that uses the Currency format.
EXACT (<i>text1</i> , <i>text2</i>)	Compares two strings to see whether they are identical.
FIND (<i>find</i> , <i>within</i> [, <i>start</i>])	Returns the character position of the text <i>find</i> within the text <i>within</i> . FIND is case sensitive.
FIXED (<i>number</i> [, <i>decimals</i>][, <i>no_commas</i>])	Converts <i>number</i> to a string that uses the Number format.
LAMBDA ([<i>parameter1</i>] [, <i>parameter2</i>]... <i>calculation</i>)	Defines a custom function that returns the result of the <i>calculation</i> using the optional parameters as inputs (see Chapter 9).
LEFT (<i>text</i> [, <i>number</i>])	Returns the leftmost <i>number</i> of characters from <i>text</i> .
LEN (<i>text</i>)	Returns the length of <i>text</i> .
LET (<i>name</i> , <i>value</i> , <i>calculation</i>)	Assigns <i>value</i> to the variable <i>name</i> and uses that variable as an input to the <i>calculation</i> , the result of which is returned by the function (see Chapter 9).
LOWER (<i>text</i>)	Converts <i>text</i> to lowercase.
MID (<i>text</i> , <i>start</i> , <i>number</i>)	Returns <i>number</i> characters from <i>text</i> , starting at <i>start</i> .
NUMBERVALUE (<i>text</i> , <i>decimal</i> , <i>group</i>)	Converts <i>text</i> to a number by interpreting the decimal symbol within <i>text</i> as the <i>decimal</i> separator and the <i>group</i> symbol as the group separator.
PROPER (<i>text</i>)	Converts <i>text</i> to proper case (first letter of each word capitalized).
REPLACE (<i>old</i> , <i>start</i> , <i>chars</i> , <i>new</i>)	Replaces the <i>old</i> string with the <i>new</i> string.
REPT (<i>text</i> , <i>number</i>)	Repeats the <i>text</i> string <i>number</i> times.
RIGHT (<i>text</i> [, <i>number</i>])	Returns the rightmost <i>number</i> characters from <i>text</i> .
SEARCH (<i>find</i> , <i>within</i> [, <i>start_num</i>])	Returns the character position of the text <i>find</i> within the text <i>within</i> . SEARCH is not case sensitive.
SUBSTITUTE (<i>text</i> , <i>old</i> , <i>new</i> [, <i>num</i>])	In <i>text</i> , substitutes the <i>new</i> string for the <i>old</i> string; optionally substitutes only the instance specified by <i>num</i> .
T (<i>value</i>)	Converts <i>value</i> to text.
TEXT (<i>value</i> , <i>format</i>)	Formats <i>value</i> and converts it to text.
TEXTJOIN (<i>delimiter</i> , <i>ignore_empty</i> , <i>text1</i> , <i>text2</i> ,...)	Joins the text strings, separating each with the <i>delimiter</i> character.
TRIM (<i>text</i>)	Removes excess spaces from <i>text</i> .
UNICHAR (<i>number</i>)	Returns the character that corresponds to the UNICODE value given by <i>number</i> .
UNICODE (<i>text</i>)	Returns the UNICODE value for the first character in <i>text</i> .
UPPER (<i>text</i>)	Converts <i>text</i> to uppercase.
VALUE (<i>text</i>)	Converts <i>text</i> to a number.
VALUETOTEXT (<i>value</i>)	Returns the text representation of <i>value</i> .

فصل 5: کارکردن با توابع متنی

کارکردن با کاراکترها و کدها

هر کاراکتری که می‌توانید روی صفحه‌نمایش دهید دارای یک کد عددی زیرین مخصوص به خود است. به‌عنوان مثال، کد حرف بزرگ A معادل 65 است، در حالی که کد کاراکتر & (امپرساند) معادل 38 است. این کدها نه تنها به کاراکترهای الفبایی-عددانی که از طریق صفحه کلید شما قابل دسترسی هستند، بلکه به کاراکترهای اضافی نیز تعلق دارند. مجموعه این کاراکترها به نام مجموعه کاراکتر ANSI شناخته می‌شود و اعداد اختصاص داده شده به هر کاراکتر به نام کدهای ANSI شناخته می‌شوند.

به‌عنوان مثال، کد ANSI برای کاراکتر حق چاپ (©) معادل 169 است. برای نمایش این کاراکتر، کلیدهای Alt+0169 را فشار داده و از شماره‌ساز صفحه کلید خود برای وارد کردن ارقام استفاده کنید.

 **یادداشت:** در هنگام وارد کردن ارقام، به یاد داشته باشید که همیشه باید صفر پیشرو را برای کدهای بالای 127 اضافه کنید.

کدهای ANSI از 1 تا 255 ادامه دارند، هرچند که 31 کد اول کدهای غیرچاپی هستند که کاراکترهایی مانند بازگشت به خط و فید خط را تعریف می‌کنند.

تابع CHAR

اکسل به شما این امکان را می‌دهد که کاراکتری که توسط یک کد ANSI نشان داده می‌شود را با استفاده از تابع CHAR تعیین کنید: CHAR(number):number است که باید عددی بین 1 و 255 باشد.

به‌عنوان مثال، فرمول زیر نماد حق چاپ (Copyright Symbol) را نمایش می‌دهد (کد ANSI 169):

=CHAR(169)

 **یادداشت:** اگر با مقادیر UNICODE به جای مقادیر ANSI کار می‌کنید، از تابع UNICHAR به جای تابع CHAR استفاده کنید.

تولید مجموعه کاراکتر ANSI

شکل 5-1 یک کاربرگ را نشان می‌دهد که کل مجموعه کاراکتر ANSI را نمایش می‌دهد، به جز 31 کاراکتر غیرچاپی اول (و توجه داشته باشید که کد ANSI 32 نمایانگر کاراکتر فاصله است). در هر مورد، کاراکتر با اعمال تابع CHAR به مقدار موجود در سلول سمت چپ نمایش داده شده است.

فصل 5: کار کردن با توابع متنی

یادداشت: کاراکتر واقعی که توسط یک کد ANSI نمایش داده می شود به فونتی که به سلول اعمال شده بستگی دارد. کاراکترهای نشان داده شده در شکل 5-1 کاراکترهایی هستند که با فونت های متن معمولی، مانند Arial، مشاهده می کنید. با این حال، اگر فونتی مانند Symbol یا Wingdings را به کاربرگ اعمال کنید، مجموعه متفاوتی از کاراکترها را خواهید دید.

یادداشت: شما می توانید با دانلود فایل Chapter05.xlsx از هر یک از سایت های محتوای همراه که در مقدمه ذکر شده، با تمام مثال های این فصل کار کنید.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Code	CHAR()	Code	CHAR()	Code	CHAR()	Code	CHAR()	Code	CHAR()	Code	CHAR()	Code	CHAR()	Code	CHAR()	Code	CHAR()
2	32		57	9	82	R	107	k	132	„	157		182	¶	207	İ	232	è
3	33	!	58	:	83	S	108	l	133	…	158	ž	183	·	208	Đ	233	é
4	34	"	59	;	84	T	109	m	134	†	159	Ÿ	184	.	209	Ň	234	ê
5	35	#	60	<	85	U	110	n	135	‡	160		185	ˆ	210	Ò	235	ë
6	36	\$	61	=	86	V	111	o	136	ˆ	161	ı	186	˘	211	Ó	236	ì
7	37	%	62	>	87	W	112	p	137	‰	162	ç	187	»	212	Ô	237	í
8	38	&	63	?	88	X	113	q	138	Š	163	€	188	¼	213	Õ	238	î
9	39	'	64	@	89	Y	114	r	139	‹	164	¤	189	½	214	Ö	239	ï
10	40	(	65	A	90	Z	115	s	140	Œ	165	¥	190	¾	215	×	240	ð
11	41	)	66	B	91	[	116	t	141		166	¦	191	¿	216	Ø	241	ñ
12	42	*	67	C	92	\	117	u	142	Ž	167	§	192	À	217	Ù	242	ò
13	43	+	68	D	93	]	118	v	143		168	¨	193	Á	218	Ú	243	ó
14	44	,	69	E	94	^	119	w	144		169	©	194	Â	219	Û	244	ô
15	45	-	70	F	95	_	120	x	145	'	170	®	195	Ã	220	Ü	245	õ
16	46	.	71	G	96	`	121	y	146	'	171	«	196	Ä	221	Ý	246	ö
17	47	/	72	H	97	a	122	z	147	"	172	»	197	Å	222	Þ	247	÷
18	48	0	73	I	98	b	123	{	148	"	173	-	198	Æ	223	ß	248	ø
19	49	1	74	J	99	c	124		149	•	174	®	199	Ç	224	à	249	ù
20	50	2	75	K	100	d	125	}	150	-	175	-	200	È	225	á	250	ú
21	51	3	76	L	101	e	126	~	151	—	176	°	201	É	226	â	251	û
22	52	4	77	M	102	f	127	¸	152	™	177	±	202	Ê	227	ã	252	ü
23	53	5	78	N	103	g	128	€	153	™	178	²	203	Ë	228	ä	253	ý
24	54	6	79	O	104	h	129		154	§	179	³	204	Ì	229	å	254	þ
25	55	7	80	P	105	i	130	,	155	›	180	´	205	Í	230	æ	255	ÿ
26	56	8	81	Q	106	j	131	f	156	œ	181	µ	206	Î	231	ç		

شکل 5-1 این کاربرگ از تابع CHAR برای نمایش هر عضو چاپی مجموعه کاراکتر ANSI استفاده می کند.

برای ساخت مجموعه کاراکتری که در شکل 5-1 نمایش داده شده است، کد ANSI و تابع CHAR را در بالای هر ستون وارد کردم و سپس به پایین پرکردم تا بقیه ستون را تولید کنم. یک روش کمتر خسته کننده (اگرچه با نمایی کمتر مفید) این است که از تابع ROW استفاده کنید که شماره ردیف سلول فعلی را برمی گرداند. فرض کنید می خواهید جدول خود را از ردیف 2 شروع کنید؛ شما می توانید هر کاراکتر ANSI را با استفاده از فرمول زیر تولید کنید:

فصل 5: کار کردن با توابع متنی

=CHAR(ROW() + 30)

تولید یک سری از حروف

دستگیره پر کردن اکسل و دستور Home > Fill > Series برای تولید یک سری از اعداد یا تاریخها بسیار مناسب است، هنگامی که به یک سری حروف) مانند a ، b ، c و غیره (نیاز دارید، اما کارایی ندارند. باین حال، می توانید از تابع CHAR در یک فرمول آرایه ای برای تولید چنین مجموعه ای استفاده کنید .

برای تولید ستونی از حروف که با a شروع می شود) که معادل ANSI 97 است)، فرمول زیر را در محلی که می خواهید سری آغاز شود وارد کنید:

=CHAR(ROW(A97))

برای تولید یک ردیف از حروف که با a شروع می شود، فرمول زیر را در محلی که می خواهید سری آغاز شود وارد کنید:

=CHAR(COLUMN(CS1))

سپس با کشیدن دستگیره پر کردن به سمت پایین (برای یک ستون) یا به سمت راست (برای یک ردیف) سری را گسترش دهید. برای حروف بزرگ (که A معادل ANSI 65 است)، با فرمول های زیر شروع کنید و از دستگیره پر کردن برای گسترش سری استفاده کنید:

=CHAR(ROW(A65))=CHAR(COLUMN(BM1))

تابع CODE

تابع CODE معکوس تابع CHAR است. به عبارت دیگر، با توجه به یک کاراکتر متنی، CODE ارزش کد ANSI آن را برمی گرداند: CODE(text) یک کاراکتر یا رشته متنی است. توجه داشته باشید که اگر یک رشته چند کاراکتری را وارد کنید، CODE کد ANSI اولین کاراکتر در رشته را برمی گرداند.

به عنوان مثال، فرمول های زیر هر دو عدد 83 را برمی گردانند، که کد ANSI حرف بزرگ S است:

CODE("S")CODE("Spacely Sprockets")

 یادداشت: اگر به شناسایی مقدار UNICODE یک کاراکتر به جای مقدار ANSI آن نیاز دارید، از تابع UNICODE به جای تابع CODE استفاده کنید.

فصل 5: کار کردن با توابع متنی

تولید یک سری از حروف با شروع از هر حرفی

در بخش‌های قبلی، یاد گرفتید که چگونه تابع CHAR و ROW را در یک فرمول ترکیب کنید تا یک سری از حروف را با شروع از حرف a یا A تولید کنید. اما اگر بخواهید با حرف دیگری شروع کنید چه؟ شما می‌توانید با تغییر مقدار اولیه که به تابع CHAR وارد می‌شود، این کار را انجام دهید. در مثال قبلی، از 97 برای شروع سری با حرف a استفاده کردم، اما می‌توانید از 98 برای شروع با b، 99 برای شروع با c و غیره استفاده کنید.

به جای جستجوی کد ANSI کاراکتر موردنظر، از تابع CODE استفاده کنید تا اکسل این کار را برای شما انجام دهد:
`CHAR(CODE("letter") + ROW(range) - ROW(first_cell))=`

در اینجا، به جای letter با حرفی که می‌خواهید سری را با آن شروع کنید، جایگزین کنید. به عنوان مثال، فرمول زیر سری را با حرف بزرگ N شروع می‌کند:

`=CHAR(CODE("N") + ROW(A1:A13) - ROW(A1))`

💡 نکته: اگر از نسخه‌ای از اکسل قبل از 2019 استفاده می‌کنید، هنگام کار با فرمول‌های این بخش، به یاد داشته باشید که آن‌ها را به عنوان فرمول‌های آرایه‌ای در محدوده مشخص شده در اولین تابع ROW (برای مثال، A1:A13) وارد کنید.

تبدیل متن

تخصص اکسل در محاسبات عددی است و اغلب به رشته‌ها توجه کمتری می‌شود، به ویژه در زمان نمایش رشته‌ها در یک کاربرگ. به عنوان مثال، الحاق یک مقدار عددی به یک رشته باعث می‌شود که عدد بدون هیچ گونه فرمت خاصی نمایش داده شود، حتی اگر سلول اصلی یک فرمت عددی داشته باشد. به همین ترتیب، رشته‌های وارد شده از یک پایگاه داده یا فایل متنی می‌توانند دارای حالت نادرست یا بدون فرمت باشند. با این حال، همان طور که در بخش‌های بعدی خواهید دید، اکسل تعداد زیادی تابع کاربرگ ارائه می‌دهد که به شما اجازه می‌دهد رشته‌ها را به یک فرمت متنی مناسب تبدیل کنید یا بین مقادیر متنی و عددی تبدیل کنید.

تابع LOWER

تابع LOWER یک رشته مشخص شده را به حروف کوچک تبدیل می‌کند: `LOWER(text)` رشته‌ای که می‌خواهید به حروف کوچک تبدیل کنید.

به عنوان مثال، فرمول زیر متن موجود در سلول B10 را به حروف کوچک تبدیل می‌کند:

فصل 5: کارکردن با توابع متنی

=LOWER(B10)

تابع LOWER اغلب برای تبدیل داده‌های واردشده، به‌ویژه داده‌هایی که از یک کامپیوتر اصلی وارد می‌شوند و معمولاً به‌صورت حروف بزرگ ارائه می‌شوند، استفاده می‌شود.

تابع UPPER

تابع UPPER یک رشته مشخص شده را به حروف بزرگ تبدیل می‌کند: UPPER(text) رشته‌ای که می‌خواهید به حروف بزرگ تبدیل کنید.

به‌عنوان مثال، فرمول زیر متن در سلول‌های A5 و B5 را به حروف بزرگ تبدیل می‌کند و نتایج را با یک فاصله بین آنها الحاق می‌کند:

=UPPER(A5) & " " & UPPER(B5)

تابع PROPER

تابع PROPER یک رشته مشخص شده را به حالت مناسب (proper case) تبدیل می‌کند، به این معنی که حرف اول هر کلمه به‌صورت بزرگ و بقیه حروف به‌صورت کوچک نمایش داده می‌شود: PROPER(text) رشته‌ای که می‌خواهید به حالت مناسب تبدیل کنید.

به‌عنوان مثال، فرمول زیر (که به‌عنوان یک آرایه از ده سلول عمودی در اکسل 2016 یا نسخه‌های قبلی وارد شده است) متن موجود در محدوده A1:A10 را به حالت مناسب تبدیل می‌کند:

=PROPER(A1:A10)

تابع NUMBERVALUE

تابع NUMBERVALUE یک مقدار متنی را به یک عدد تبدیل می‌کند و نماد مورد استفاده برای اعشار و گروه‌بندی را مشخص می‌کند:

NUMBERVALUE(text [, decimal_separator] [, group_separator])

- text نمایندگی متنی از عدد است.
- decimal_separator نمادی که در متن برای جداسازی بخش صحیح از بخش کسری استفاده می‌شود.

فصل 5: کار کردن با توابع متنی

- **group_separator** نمادی که در متن برای جداسازی گروه‌های عددی، مانند هزار و میلیون، استفاده می‌شود.

این تابع زمانی مفید است که با یک کاربرگ مواجه شوید که مانند اعداد با نمادهای غیرمعیار برای جداکننده‌های اعشار و گروه‌بندی استفاده می‌شود.

برای مثال، فرض کنید سلول C6 حاوی مقدار 12,34 است، که در آن کاما (,) به‌عنوان جداساز اعشار استفاده می‌شود (که در بسیاری از مناطق اروپایی متداول است). برای تبدیل این مقدار به یک عدد که از نقطه استفاده می‌کند (با فرض اینکه این نماد جداساز اعشار محلی شماست)، می‌توانید از فرمول زیر استفاده کنید:

=NUMBERVALUE(C6, ",")

به‌عنوان مثالی دیگر، فرض کنید سلول E5 حاوی مقدار 2~345'67 است، که در آن تیلت (~) به‌عنوان جداساز گروه (هزارها در این مورد) و آپستروف (') به‌عنوان جداساز اعشار عمل می‌کند. در این صورت، فرمول زیر این مقدار را به یک عدد تبدیل می‌کند:

=NUMBERVALUE(E5, "'", "~")

تابع ARRAYTOTEXT

تابع ARRAYTOTEXT یک محدوده از مقادیر را به‌عنوان نمایندگی متنی برمی‌گرداند:

ARRAYTOTEXT(array [, format])

- **array** محدوده‌ای است که می‌خواهید به‌عنوان متن برگردانید.
- **format** فرمت متنی بازگشتی: 0 (پیش‌فرض) متن را به‌عنوان یک فهرست مختصر با هر مورد جدا شده توسط کاما و فضای خالی باز می‌گرداند؛ 1 متن را با استفاده از یک فرمت استرکت آرایه‌ای باز می‌گرداند، به طوری که ردیف‌ها با نقطه‌ویرگول جدا شده و مقادیر متنی در علامت نقل قول قرار می‌گیرند.

آرگومان فرمت به‌صورت مختصر (0) و استرکت (1) کمی گیج‌کننده است، بنابراین بیایید یک مثال ساده را بررسی کنیم. فرض کنید مقادیر 123، Primrose Lane و Indianapolis در سلول‌های A1، B1 و C1 به ترتیب و مقادیر 987، Blue Jay Way و Toronto در سلول‌های A2، B2 و C2 قرار دارند. فرمول زیر:

=ARRAYTOTEXT(A1:C2, 0)

فصل 5: کارکردن با توابع متنی

متن زیر را برمی گرداند:

Primrose Lane, Indianapolis, 987, Blue Jay Way, Toronto, 123

اما فرمول زیر:

=ARRAYTOTEXT(A1:C2, 1)

متن زیر را برمی گرداند:

{"Primrose Lane","Indianapolis";987,"Blue Jay Way","Toronto",123}

به گروه‌های بزرگ که اکنون در اطراف متن ظاهر می‌شوند دقت کنید، که نشان می‌دهد نتیجه یک آرایه است؛ همچنین به نقطه‌ویرگول (;) بین "Indianapolis" و 987 توجه کنید که نشان می‌دهد آرایه یک ردیف جدید را شروع می‌کند.

تابع VALUETOTEXT

تابع VALUETOTEXT یک نمایندگی متنی از یک مقدار را بازمی گرداند:

VALUETOTEXT(value[, format])

value مقداری است که می‌خواهید به عنوان متن برگردانید (که می‌تواند یک محدوده از مقادیر باشد).

format فرمت متنی بازگشتی: 0 (پیش فرض) متن را به عنوان یک فهرست مختصر با هر مورد جدا شده توسط کاما و فضای خالی بازمی گرداند؛ 1 متن را با استفاده از یک فرمت استرکت آرایه‌ای بازمی گرداند، به طوری که ردیف‌ها با نقطه‌ویرگول جدا شده و مقادیر متنی در علامت نقل قول قرار می‌گیرند.

شاید رایج‌ترین استفاده از VALUETOTEXT تبدیل یک مقدار سلول که از یکی از نوع‌های داده پیشرفته اکسل - مانند جغرافیا یا سهام - استفاده می‌کند به یک نمایندگی متنی باشد. برای مثال، فرض کنید سلول A1 حاوی Indianapolis با فرمت نوع داده جغرافیا است. برای دریافت فقط نام "Indianapolis" از آن ورودی، معمولاً از فرمول زیر استفاده می‌کنید:

=VALUETOTEXT(A1)

فرمت‌بندی متن

در فصل 1 یاد گرفتید که می‌توانید با استفاده از فرمت‌های عددی داخلی یا سفارشی، نتایج فرمول‌های خود را با کنترل مواردی مانند کاما، اعداد اعشاری، نمادهای ارزی و موارد دیگر بهبود ببخشید. این برای نتایج سلول‌ها خوب است، اما اگر بخواهید نتیجه‌ای را درون یک رشته بگنجانید چه؟ مثلاً به فرمول متنی زیر توجه کنید:

فصل 5: کارکردن با توابع متنی

= "The expense total for this quarter in 2022 is " & F11

صرف نظر از اینکه شما نتیجه را در سلول F11 چگونه فرمت کرده‌اید، عدد درون رشته به صورت فرمت عددی عمومی اکسل نمایش داده می‌شود. به عنوان مثال، اگر سلول F11 مقدار \$74,400 را شامل شود، فرمول قبلی در سلول به صورت زیر ظاهر خواهد شد:

The expense total for this quarter in 2022 is 74400

شما به روشی نیاز دارید تا عدد را درون رشته فرمت کنید. سه بخش بعدی توابع اکسل را به شما نشان می‌دهند که به شما اجازه می‌دهند دقیقاً همین کار را انجام دهید.

در ادامه با توابعی که برای فرمت بندی اعداد در متن استفاده می‌شوند، آشنا می‌شوید.

تابع DOLLAR

تابع DOLLAR یک مقدار عددی را به یک مقدار متنی که از فرمت ارزی استفاده می‌کند، تبدیل می‌کند:

DOLLAR(number [, decimals])

number عددی است که می‌خواهید تبدیل کنید.

decimals تعداد ارقام اعشاری را که می‌خواهید نمایش دهید مشخص می‌کند (پیش فرض 2 است).

برای رفع مثال رشته‌ای از بخش قبلی، شما نیاز دارید تابع DOLLAR را بر روی سلول F13 اعمال کنید:

= "The expense total for this quarter in 2022 is " & DOLLAR(F13, 0)

در این حالت، عدد بدون ارقام اعشاری فرمت بندی می‌شود. به عنوان مثال، اگر مقدار موجود در سلول F13 برابر با 74400 باشد، این فرمول نتیجه را به صورت:

The expense total for this quarter in 2022 is \$74,400

نمایش می‌دهد. شکل 5-2 یک تغییر در این فرمول را در سلول B16 به تصویر می‌کشد. (برای مقایسه، سلول B15 شامل یک فرمول مشابه است که از تابع DOLLAR استفاده نمی‌کند.)

فصل 5: کار کردن با توابع متنی

B16 fx ="The expense total for this quarter in 2022 is " & DOLLAR(F13,0)									
	A	B	C	D	E	F	G	H	I
1	Expense Budget Calculation - 1st Quarter								
2									
3		INCREASE	1.03						
4									
5		EXPENSES	January	February	March	Total			
6		Advertising	\$4,600	\$4,200	\$5,200	\$14,000			
7		Rent	\$2,100	\$2,100	\$2,100	\$6,300			
8		Supplies	\$1,300	\$1,200	\$1,400	\$3,900			
9		Salaries	\$16,000	\$16,000	\$16,500	\$48,500			
10		Utilities	\$500	\$600	\$600	\$1,700			
11		2021 TOTAL	\$24,500	\$24,100	\$25,800	\$74,400			
12									
13		2022 BUDGET	\$25,235	\$24,823	\$26,574	\$76,632			
14									
15		The expense total for this quarter in 2021 is 74400							
16		The expense total for this quarter in 2022 is \$76,632							
17									

شکل 5-2: از تابع DOLLAR برای نمایش یک عدد به عنوان یک رشته با فرمت ارزی استفاده کنید.

تابع FIXED

برای برخی از انواع اعداد، می توانید تعداد اعشار و اینکه آیا کاما به عنوان جدا ساز هزارگان وارد می شود را با استفاده از تابع FIXED کنترل کنید:

FIXED(number [, decimals] [, no_commas])

number عددی است که می خواهید به رشته تبدیل کنید.

decimals تعداد اعشار را که باید نمایش داده شود مشخص می کند (پیش فرض 2 است).

no_commas یک مقدار منطقی است که تعیین می کند آیا کاماها به رشته وارد می شوند یا خیر. از TRUE برای حذف کاماها و از FALSE برای درج کاماها استفاده کنید (این پیش فرض است).

به عنوان مثال، فرمول زیر از تابع SUM برای جمع یک محدوده استفاده می کند و تابع FIXED را به نتیجه اعمال می کند تا به عنوان یک رشته با کاماها و بدون ارقام اعشاری نمایش داده شود:

فصل 5: کارکردن با توابع متنی

= "Total show attendance: " & FIXED(SUM(A1:A8), 0, FALSE) & " people".

تابع TEXT

تابع TEXT ابزاری قدرتمند است که به شما کنترل کاملی بر نحوه فرمت یک عدد، تاریخ، یا زمان در داخل یک رشته می‌دهد. اگر بخواهید فرمت عددی را به طور دلخواه تنظیم کنید یا تاریخ‌ها و زمان‌ها را در رشته‌ها بگنجانید، تابع TEXT مناسب است:

TEXT(number, format)

number عدد، تاریخ یا زمان موردنظر برای تبدیل است.

format فرمت عددی یا تاریخ/زمانی است که می‌خواهید به عدد اعمال کنید.

قدرت تابع TEXT در آرگومان فرمت آن نهفته است که یک فرمت سفارشی است که مشخص می‌کند دقیقاً می‌خواهید عدد به چه شکلی نمایش داده شود.

جدول 2-5 نمادهای خاصی را لیست می‌کند که می‌توانید برای تعریف یک فرمت عددی استفاده کنید.

جدول 2-5: نمادهای فرمت‌بندی عددی

Symbol	Description
General	Displays the number with the General format.
#	Holds a place for a digit and displays the digit exactly as typed. Displays nothing if no number is entered.
0	Holds a place for a digit and displays the digit exactly as typed. Displays 0 if no number is entered.
?	Holds a place for a digit and displays the digit exactly as typed. Displays a space if no number is entered.
. (period)	Sets the location of the decimal point.
, (comma)	Sets the location of the thousands separator. Marks only the location of the first thousand.
%	Multiplies the number by 100 (for display only) and adds the percent (%) character.
E+ e+ E- e-	Displays the number in scientific format. E- and e- place a minus sign in the exponent; E+ and e+ place a plus sign in the exponent.
/ (slash)	Sets the location of the fraction separator.
\$ () : - + <space>	Displays the character.
*	Repeats whatever character immediately follows the asterisk until the cell is full. Doesn't replace other symbols or numbers.
_ (underscore)	Inserts a blank space the width of whatever character follows the underscore.
\ (backslash)	Inserts the character that follows the backslash.
"text"	Inserts the text that appears within the quotation marks.
@	Holds a place for text.

فصل 5: کارکردن با توابع متنی

جدول 3-5: نمادهای فرمت‌بندی تاریخ و زمان

Symbol	Description
<i>Date Formats</i>	
d	Day number without a leading zero (1—31)
dd	Day number with a leading zero (01—31)
ddd	Three-letter day abbreviation (Mon , for example)
dddd	Full day name (Monday , for example)
m	Month number without a leading zero (1—12)
mm	Month number with a leading zero (01—12)
mmm	Three-letter month abbreviation (Aug , for example)

mmmm	Full month name (August , for example)
yy	Two-digit year (00—99)
yyyy	Full year (1900—9999)
<i>Time Formats</i>	
h	Hour without a leading zero (0—24)
hh	Hour with a leading zero (00—24)
m	Minute without a leading zero (0—59)
mm	Minute with a leading zero (00—59)
s	Second without a leading zero (0—59)
ss	Second with a leading zero (00—59)
AM/PM, am/pm, A/P	12-hour clock time
/ : . —	Symbols used to separate parts of dates or times

به‌عنوان مثال، فرمول زیر از تابع AVERAGE برای محاسبه میانگین محدوده A1:A31 استفاده می‌کند و سپس از تابع TEXT برای اعمال فرمت سفارشی #,##0.00°F به نتیجه استفاده می‌کند:

"=The average temperature was " & TEXT(AVERAGE(A1:A31), "#,##0.00°F")

✎ یادداشت: برای درج نماد درجه (°)، با استفاده از کیبورد خود، کلیدهای Alt+0176 را فشار دهید.

نمایش زمان آخرین به‌روزرسانی یک کتاب کار

بسیاری از افراد دوست دارند کتاب‌های کار خود را با تنظیم حالت محاسبه دستی اکسل و واردکردن تابع NOW در یک سلول، anotate کنند (که تاریخ و زمان فعلی را برمی‌گرداند). تابع NOW به‌روزرسانی نمی‌شود مگر اینکه شیت را ذخیره کنید یا محاسبات را انجام دهید، بنابراین همیشه می‌دانید که شیت آخرین بار چه زمانی به‌روزرسانی شده است.

فصل 5: کارکردن با توابع متنی

به جای ورود ساده NOW، ممکن است بهتر باشد تاریخ را با یک رشته توضیحی آغاز کنید، مانند: "This workbook last updated: ". برای انجام این کار، می توانید فرمول زیر را وارد کنید:

= "This workbook last updated: " & NOW

متأسفانه، خروجی شما به شکل زیر خواهد بود:

This workbook last updated: 42238.51001

عدد 42238.51001 نمای داخلی اکسل از تاریخ و زمان است. (عدد سمت چپ اعشار نشان دهنده تاریخ و عدد سمت راست اعشار نشان دهنده زمان است.) برای دریافت یک تاریخ و زمان به درستی فرمت بندی شده، از تابع TEXT استفاده کنید. به عنوان مثال، برای فرمت دادن نتایج تابع NOW به فرمت MM/DD/YY HH:MM، از فرمول زیر استفاده کنید:

= "This workbook last updated: " & TEXT(NOW, "mm/dd/yy hh:mm")

دست کاری متن

این بخش شما را با حقایق اصلی ترفندهای دست کاری متن در اکسل آشنا می کند. تمام توابعی که در صفحات آینده یاد خواهید گرفت مفید خواهند بود، اما خواهید دید که با ترکیب دو یا چندتا از این توابع در یک فرمول واحد، می توانید انعطاف پذیری شگفت انگیز توانایی های دست کاری متن اکسل را به نمایش بگذارید.

حذف کاراکترهای نامطلوب از یک رشته

کاراکترهایی که از پایگاه های داده و فایل های متنی وارد می شوند، معمولاً با انواع مختلفی از کاراکترهای اضافی همراه هستند که نیازی به آنها ندارید. این کاراکترها می توانند فضایی اضافی در رشته باشند یا شامل تغذیه های خط، بازگشت های کاراکتر و سایر کاراکترهای غیرقابل چاپ درون رشته باشند. برای حل این مشکلات، اکسل چندین تابع را ارائه می دهد: TRIM و CLEAN.

تابع TRIM

شما از تابع TRIM برای حذف فضاهای اضافی درون یک رشته استفاده می کنید:

TRIM(text)

text رشته ای است که می خواهید فضاهای اضافی از آن حذف شوند.

در اینجا، "اضافی" به معنای همه فضاهای قبل و بعد از رشته و همچنین دو یا چند فضای متوالی درون رشته است. در مورد اخیر، TRIM همه فضاهای متوالی را به جز یکی حذف می کند.

فصل 5: کار کردن با توابع متنی

شکل 3-5 نشان‌دهنده عملکرد تابع TRIM است. هر رشته در محدوده A2:A7 شامل تعدادی فضای اضافی قبل، داخل یا بعد از نام است. توابع TRIM در ستون C ظاهر می‌شوند. برای کمک به تأیید عملکرد تابع TRIM، من از تابع LEN در ستون‌های B و D استفاده می‌کنم. تابع LEN تعداد کاراکترهای یک رشته مشخص را برمی‌گرداند و از نحوه استفاده زیر پیروی می‌کند:

LEN(text)

text رشته‌ای است که می‌خواهید تعداد کاراکترهای آن را بدانید.

	A	B	C	D	E
1	Original String	Length	Trimmed String	Length	
2	Maria Anders	16	Maria Anders	12	
3	Ana Trujillo	17	Ana Trujillo	12	
4	Antonio Moreno	23	Antonio Moreno	14	
5	Thomas Hardy	17	Thomas Hardy	12	
6	Angus Glen Dunlop	26	Angus Glen Dunlop	17	
7	Christina Berglund	22	Christina Berglund	18	

شکل 3-5: استفاده از تابع TRIM برای حذف فضاهای اضافی از یک رشته

در این شکل، مشاهده می‌کنید که چگونه تابع TRIM به کار می‌رود تا فضاهای اضافی از رشته‌های موجود در محدوده A2:A7 حذف شود. به علاوه، ستون‌های مربوط به تابع LEN در سمت چپ و راست به تأیید تعداد کاراکترهای رشته‌های قبل و بعد از استفاده از تابع TRIM کمک می‌کند. این تجزیه و تحلیل به شما نشان می‌دهد که چگونه این تابع می‌تواند به بهبود کیفیت داده‌ها کمک کند و رشته‌ها را تمیز کند.

تابع CLEAN

شما از تابع CLEAN برای حذف کاراکترهای غیرقابل چاپ از یک رشته استفاده می‌کنید:

CLEAN(text)

text رشته‌ای است که می‌خواهید کاراکترهای غیرقابل چاپ از آن حذف شوند.

به یاد داشته باشید که کاراکترهای غیرقابل چاپ شامل کدهای 1 تا 31 از مجموعه کاراکتر ANSI هستند. تابع CLEAN بیشتر برای حذف تغذیه‌های خط (ANSI 10) یا بازگشت‌های کاراکتر (ANSI 13) از داده‌های چندخطی استفاده می‌شود.

فصل 5: کار کردن با توابع متنی

شکل 4-5 یک مثال از استفاده از تابع CLEAN را نمایش می‌دهد. در این شکل، مشاهده می‌کنید که چگونه این تابع می‌تواند کاراکترهای غیرضروری را از یک رشته حذف کند و دقت داده‌ها را افزایش دهد.

B2		
fx =CLEAN(A2)		
A	B	C
1	Original String	Cleaned String
	49 Gilbert St. London, England	
2	EC1 4SD	49 Gilbert St.London, England EC1 4SD
	P.O. Box 78934 New Orleans, LA	
3	70117	P.O. Box 78934New Orleans, LA 70117
	707 Oxford Rd. Ann Arbor, MI	
4	48104	707 Oxford Rd.Ann Arbor, MI 48104
5		

شکل 4-5: استفاده از تابع CLEAN برای حذف کاراکترهای غیرقابل چاپ مانند تغذیه‌های خط از یک رشته

در این شکل، نشان داده شده است که چگونه تابع CLEAN به کار می‌رود تا کاراکترهای غیرقابل چاپ، از جمله تغذیه‌های خط، را از رشته‌های موجود در داده‌ها حذف کند. این عملکرد به تمیز کردن داده‌ها و بهبود کیفیت ورودی‌ها کمک می‌کند و فضایی مناسب برای تجزیه و تحلیل اطلاعات فراهم می‌آورد.

تابع TEXTJOIN: الحاق متن با یک جداکننده

در فصل 1، به اپراتور الحاق اکسل (&) اشاره کردم که برای پیوستن دو یا چند رشته به یک مقدار متنی واحد استفاده می‌شود. این اپراتور عملکرد خوبی دارد، اما زمانی که بخواهید رشته‌های الحاق شده را با یک یا چند کاراکتر جدا کنید، کار کمی دشوار می‌شود. به عنوان مثال، فرض کنید چهار ستون از داده‌های تماس دارید: عنوان، نام، حرف وسط و نام خانوادگی هر شخص. هنگامی که این مقادیر را الحاق می‌کنید، می‌خواهید فاصله‌ای بین هر یک ایجاد کنید، بنابراین ترکیب تمام چهار ستون به یک مقدار متنی واحد نیاز به فرمولی مانند زیر دارد:

=A2 & " " & B2 & " " & C2 & " " & D2

این کمی پیچیده‌ای به نظر می‌رسد و اگر هر کدام از تماس‌ها داده‌ای خالی (مانند نبود عنوان یا حرف وسط) داشته باشد، وضعیت بدتر می‌شود. برای مدیریت بهتر چنین الحاق رشته‌ای، اکسل اکنون تابع TEXTJOIN را ارائه می‌دهد:

TEXTJOIN(delimiter, ignore_empty, text1, text2, ...)

فصل 5: کار کردن با توابع متنی

delimiter کاراکتری است که می‌خواهید بین هر رشته وارد کنید.

ignore_empty یک مقدار منطقی است؛ برای عدم شامل کردن سلول‌های خالی در الحاق، از TRUE استفاده کنید (این پیش‌فرض است)؛ برای شامل کردن سلول‌های خالی، از FALSE استفاده کنید.

text1, text2, ... رشته‌هایی هستند که می‌خواهید آن‌ها را الحاق کنید.

شکل 5-5 یک مثال از استفاده تابع TEXTJOIN را نمایش می‌دهد و نشان می‌دهد که چگونه می‌توان با این تابع الحاق رشته‌ها را به راحتی و با دقت انجام داد.

E2 fx =TEXTJOIN(" ", TRUE, A2, B2, C2, D2)					
	A	B	C	D	E
1	Title	First Name	Initial	Last Name	Full Name
2	Ms.	Maria	S.	Anders	Ms. Maria S. Anders
3	Ms.	Ana	P.	Trujillo	Ms. Ana P. Trujillo
4	Mr.	Antonio		Moreno	Mr. Antonio Moreno
5	Dr.	Thomas	K.	Hardy	Dr. Thomas K. Hardy
6	Prof.	Christina	N.	Berglund	Prof. Christina N. Berglund
7	Ms.	Hanna	J.	Moos	Ms. Hanna J. Moos
8	Mr.	Frédérique	P.	Citeaux	Mr. Frédérique P. Citeaux
9	Mr.	Martín		Sommer	Mr. Martín Sommer
10	Mr.	Laurence	M.	Lebihan	Mr. Laurence M. Lebihan
11	Dr.	Elizabeth	C.	Lincoln	Dr. Elizabeth C. Lincoln
12	Ms.	Victoria	R.	Ashworth	Ms. Victoria R. Ashworth
13	Dr.	Patricio	V.	Simpson	Dr. Patricio V. Simpson
14	Mr.	Francisco	Z.	Chang	Mr. Francisco Z. Chang
15		Yang	F.	Wang	Yang F. Wang
16	Mr.	Pedro	Z.	Afonso	Mr. Pedro Z. Afonso
17	Prof.	Elizabeth	Z.	Brown	Prof. Elizabeth Z. Brown

شکل 5-5: استفاده از تابع TEXTJOIN برای الحاق چندین رشته با یک کاراکتر جداکننده بین آن‌ها

در این شکل، نشان داده شده است که چگونه تابع TEXTJOIN برای الحاق چندین رشته با استفاده از یک کاراکتر جداکننده به کار می‌رود. این تابع به راحتی مدیریت رشته‌ها را ساده‌تر می‌کند و به شما این امکان را می‌دهد که در صورت لزوم از ظرفیت حذف داده‌های خالی بهره‌مند شوید، بدون اینکه نیاز به نوشتن فرمول‌های پیچیده داشته باشید.

تابع REPT: تکرار یک کاراکتر یا رشته

تابع REPT یک کاراکتر یا رشته را به تعداد مشخصی تکرار می‌کند: REPT(text, number)

فصل 5: کار کردن با توابع متنی

text کاراکتر یا رشته ای است که می خواهید تکرار شود.

number تعداد دفعاتی است که می خواهید متن تکرار شود.

پر کردن یک سلول

تابع REPT گاهی اوقات برای پر کردن یک سلول با کاراکترها استفاده می شود. به عنوان مثال، می توانید از آن برای اضافه کردن نقاط پیشرو یا پسرو در یک سلول استفاده کنید. در اینجا یک فرمول وجود دارد که نقاط پس رو را بعد از یک رشته ایجاد می کند:

```
= "Advertising" & REPT(".", 20 - LEN("Advertising"))
```

این فرمول رشته Advertising را می نویسد و سپس از REPT برای تکرار کاراکتر نقطه بر اساس عبارت زیر استفاده می کند: 20 - LEN("Advertising"). این عبارت اطمینان می دهد که تعداد ثابتی از کاراکترها در سلول نوشته می شود. از آنجا که "Advertising" 11 کاراکتر دارد، نتیجه عبارت 9 خواهد بود که به این معنی است که نه نقطه به سمت راست رشته اضافه می شود. اگر رشته Rent (چهار کاراکتر) باشد، 16 نقطه به عنوان پرکننده اضافه خواهد شد.

شکل 5-6 نشان می دهد که چگونه این تکنیک اثر دنباله دار نقطه را ایجاد می کند.

💡 نکته: برای بهترین نتایج، لازم است که سلولها با یک فونت مونوتایپ، مانند Courier New فرمت بندی شوند. این اطمینان می دهد که تمام کاراکترها عرض یکسانی دارند که نتیجه ای یکسان و منظم در تمام سلولها به شما می دهد.

A2		= "Advertising" & REPT(".", 20 - LEN("Advertising"))			
	A	B	C	D	E
1	EXPENSES				
2	Advertising.....				
3	Rent.....				
4	Supplies.....				
5	Salaries.....				
6	Utilities.....				

شکل 5-6: استفاده از تابع REPT برای پر کردن یک سلول با کاراکترها، مانند نقاط دنباله دار نشان داده شده در اینجا

فصل 5: کارکردن با توابع متنی

در این شکل، مشاهده می‌شود که چگونه تابع REPT برای افزودن نقاط پسرو به یک رشته استفاده می‌شود. این روش به شما اجازه می‌دهد که سلول را به صورت بصری جذاب تر و منظم تر کنید، به ویژه زمانی که از یک فونت مونوتایپ استفاده می‌کنید.

ایجاد نمودارهای متنی

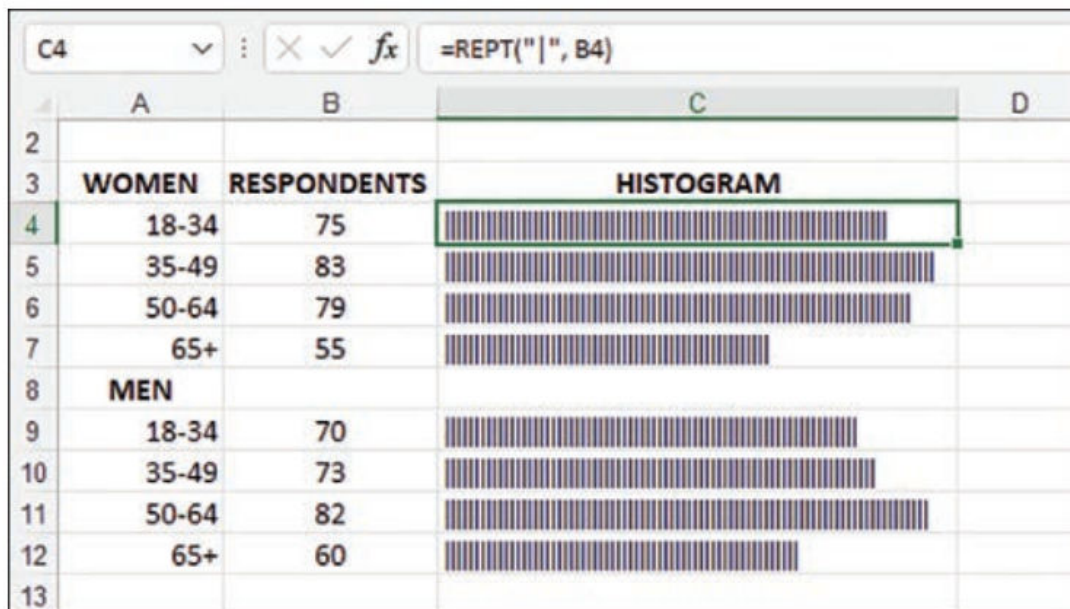
یک استفاده رایج از تابع REPT ایجاد نمودارهای مبتنی بر متن است. در این حالت، شما از یک نتیجه عددی در یک سلول به عنوان آرگومان عددی تابع REPT استفاده می‌کنید و کاراکتر تکرار شده سپس نتیجه را به نمودار تبدیل می‌کند.

یک مثال ساده از این موضوع یک هیستوگرام پایه است که فراوانی یک نمونه را در یک بازه نشان می‌دهد.

شکل 5-7 یک هیستوگرام متنی را نمایش می‌دهد که در آن بازه‌ها در ستون A و فراوانی‌ها در ستون B قرار دارند. تابع REPT در ستون C هیستوگرام را با تکرار کاراکتر عمودی (|) بر اساس هر فراوانی ایجاد می‌کند، به‌عنوان مثال با استفاده از فرمول زیر:

=REPT("|", B4)

این فرمول به راحتی نشان دهنده میزان فراوانی در بازه های مختلف است و به بیننده امکان می دهد تغییرات را به طور بصری مشاهده کند.



شکل 5-7: استفاده از تابع REPT برای ایجاد یک هیستوگرام مبتنی بر متن

فصل 5: کار کردن با توابع متنی

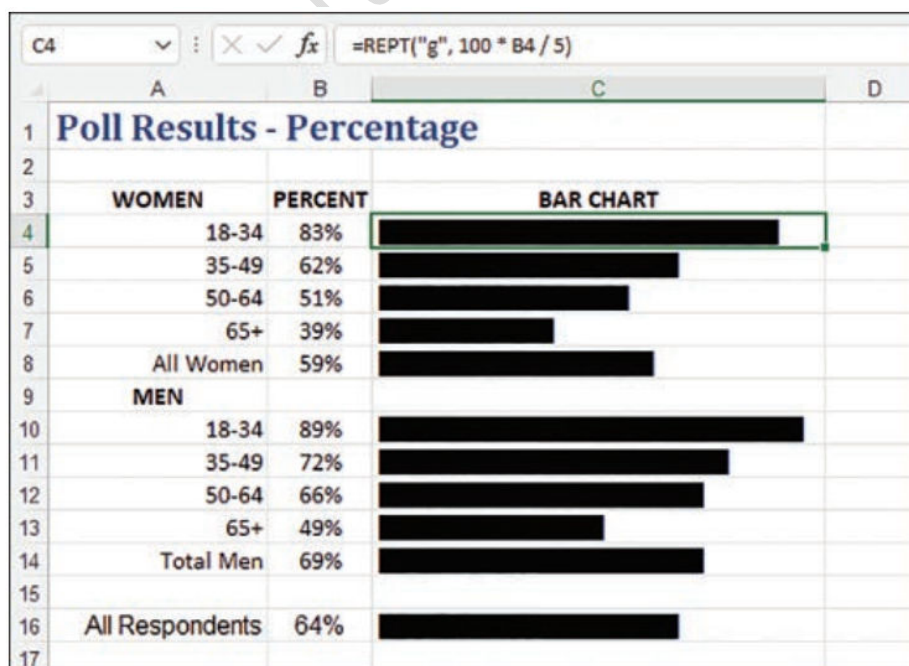
در این شکل، مشاهده می‌شود که چگونه تابع REPT برای ایجاد یک هیستوگرام متنی استفاده می‌شود. در این هیستوگرام، بازه‌ها در ستون A و فراوانی‌ها در ستون B قرار دارند، درحالی‌که ستون C نمایانگر هیستوگرام است که با استفاده از کاراکتر عمودی (|) به تعداد فراوانی‌های موجود ساخته شده است. این نحوه نمایش داده‌ها به کاربران این امکان را می‌دهد که به راحتی الگوها و توزیع‌ها را شناسایی کنند.

تبدیل هیستوگرام به یک نمودار میله‌ای مبتنی بر متن

با یک ترفند ساده، می‌توانید هیستوگرام را به یک نمودار میله‌ای مبتنی بر متن تبدیل کنید، همان‌طور که در شکل 5-8 نشان داده شده است.

ترفند در اینجا این است که سلول‌های نمودار را با فونت Webdings فرمت‌بندی کنید. در این فونت، حرف g به وسیله یک کاراکتر بلوکی نمایش داده می‌شود و تکرار آن کاراکتر یک نوار جامد تولید می‌کند. این روش به شما این امکان را می‌دهد که یک نمودار میله‌ای بصری و جذاب بسازید که در آن بارها به وضوح نشان داده می‌شوند و به راحتی داده‌ها را به تصویر می‌کشد.

نکته: برای به دست آوردن مقدار تکرار، من درصدها در ستون B را در 100 ضرب کردم تا یک عدد صحیح به دست آورم. برای حفظ کوتاهی نسبی نوارها، نتیجه را بر 5 تقسیم کردم. این روش به شما این امکان را می‌دهد که میله‌ها به صورت بصری جذاب و قابل مدیریت باشند و از شلوغی بصری جلوگیری کنند.



شکل 5-8: استفاده از تابع REPT برای ایجاد یک نمودار میله‌ای مبتنی بر متن

فصل 5: کار کردن با توابع متنی

در این شکل، نشان داده شده است که چگونه تابع REPT برای ایجاد یک نمودار میله‌ای مبتنی بر متن استفاده می‌شود. با استفاده از فونت Webdings و تکرار کاراکترهای خاص، این نمودار میله‌ای به وضوح توزیع داده‌ها را نمایش می‌دهد. نوارهای جامد نمایانگر مقادیر مختلف هستند و به کاربر اجازه می‌دهند به راحتی مقایسه‌های بصری را انجام دهد. این تکنیک راهی مؤثر برای نمایش اطلاعات است که هم از نظر طراحی و هم از نظر کاربردی جذاب است.

استفاده از نوارهای داده در اکسل

اکسل یک ویژگی به نام نوارهای داده (Data Bars) ارائه می‌دهد که به شما این امکان را می‌دهد که به سادگی تحلیل‌هایی مشابه هیستوگرام را به ورق‌های کاری خود اضافه کنید، بدون نیاز به نوشتن فرمول. برای استفاده از این ویژگی، می‌توانید به مسیر زیر بروید:

Home > Conditional Formatting > Data Bars

این قابلیت به طور خودکار نوارهای بصری را بر اساس مقادیر موجود در سلول‌ها ایجاد می‌کند و به شما این امکان را می‌دهد تا به سرعت الگوها و روندهای موجود در داده‌ها را شناسایی کنید. نوارهای داده نه تنها اطلاعات را به صورت بصری جذاب‌تر می‌کنند، بلکه فهم داده‌ها را نیز آسان‌تر می‌سازند.

استخراج یک زیررشته

ارزش‌های رشته‌ای اغلب شامل زیررشته‌های کوچک‌تر هستند که نیاز به کار با آن‌ها داریم. به عنوان مثال، در یک ستون حاوی نام‌های کامل، ممکن است بخواهید فقط به نام‌های خانوادگی پرداخته تا بتوانید داده‌ها را مرتب کنید. به طور مشابه، ممکن است بخواهید چند حرف اول از نام یک شرکت را برای گنجاندن در یک شماره حساب مرتبط با آن شرکت استخراج کنید.

اکسل به شما سه تابع برای استخراج زیررشته‌ها ارائه می‌دهد که در سه بخش بعدی توضیح داده خواهد شد.

تابع LEFT

تابع LEFT تعداد مشخصی از کاراکترها را از سمت چپ یک رشته بازمی‌گرداند:

LEFT(text [, num_chars])

text: رشته‌ای است که می‌خواهید زیررشته را از آن استخراج کنید.

num_chars: تعداد کاراکترهایی است که می‌خواهید از سمت چپ استخراج کنید (پیش فرض 1 است).

به عنوان مثال، فرمول زیر زیررشته Karen را برمی‌گرداند:

فصل 5: کارکردن با توابع متنی

=LEFT("Karen Elizabeth McHammond", 5)

این فرمول با استفاده از تابع LEFT، 5 کاراکتر اول رشته "Karen Elizabeth McHammond" را استخراج می کند.

تابع RIGHT

تابع RIGHT تعداد مشخصی از کاراکترها را از سمت راست یک رشته بازمی گرداند:

RIGHT(text [, num_chars])

text: رشته ای است که می خواهید زیررشته را از آن استخراج کنید.

num_chars: تعداد کاراکترهایی است که می خواهید از سمت راست استخراج کنید (پیش فرض 1 است).

به عنوان مثال، فرمول زیر زیررشته McHammond را برمی گرداند:

=RIGHT("Karen Elizabeth McHammond", 9)

تابع MID

تابع MID تعداد مشخصی از کاراکترها را از یک نقطه دلخواه در یک رشته بازمی گرداند:

MID(text, start_num, num_chars)

text: رشته ای است که می خواهید زیررشته را از آن استخراج کنید.

start_num: موقعیت کاراکتر که می خواهید استخراج را از آن شروع کنید.

num_chars: تعداد کاراکترهایی است که می خواهید استخراج کنید.

به عنوان مثال، فرمول زیر زیررشته Elizabeth را برمی گرداند:

=MID("Karen Elizabeth McHammond", 7, 9)

تبدیل متن به حالت جمله ای

دستور Change Case در مایکروسافت ورد گزینه ای به نام حالت جمله ای دارد که یک رشته را به حروف کوچک تبدیل می کند، به جز حرف اول که به حروف بزرگ تغییر می یابد (به همان صورتی که حروف در یک جمله عادی ظاهر می شوند).

فصل 5: کار کردن با توابع متنی

درحالی که اکسل توابع LOWER، UPPER و PROPER را دارد، اما توابعی برای تولید مستقیم حالت جمله ای ندارد. بالاین حال، می توانید یک فرمول بسازید که این کار را با استفاده از توابع LOWER و UPPER به همراه توابع LEFT و RIGHT انجام دهید.

با استخراج حرف سمت چپ و تبدیل آن به حروف بزرگ شروع می کنید (فرض بر این است که رشته در سلول A1 قرار دارد):

UPPER(LEFT(A1))

سپس همه چیز به جز حرف اول را از راست استخراج کرده و به حروف کوچک تبدیل می کنید:

LOWER(RIGHT(A1, LEN(A1) - 1))

با ترکیب این دو بخش، می توانید تبدیل به حالت جمله ای را به سادگی انجام دهید.

ترکیب نهایی برای تبدیل به حالت جمله ای

در نهایت، شما این دو عبارت را با هم ترکیب کرده و فرمول کامل زیر را به دست می آورید:

=UPPER(LEFT(A1)) & LOWER(RIGHT(A1, LEN(A1) - 1))

این فرمول ابتدا حرف اول را به حالت بزرگ و سپس باقی مانده رشته را به حالت کوچک تبدیل می کند و آن ها را به هم متصل می کند.

شکل 5-9 یک صفحه اکسل را نشان می دهد که این فرمول را به کار می گیرد و عملکرد آن را به نمایش می گذارد. این روش به شما امکان می دهد که متن ها را به فرم جملات رسمی تر تبدیل کنید و اطمینان حاصل کنید که تنها حرف اول جمله بزرگ است.

A2	=UPPER(LEFT(A1)) & LOWER(RIGHT(A1, LEN(A1) - 1))
1	IT WAS THE BEST OF TIMES, IT WAS THE WORST OF TIMES.
2	It was the best of times, it was the worst of times.
3	
4	iT wAs A dArK aNd StOrMy NiGhT.
5	It was a dark and stormy night.
6	
7	Alice Was Beginning To Get Very Tired Of Sitting By Her Sister On The Bank, And Of Having Nothing To Do.
8	Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do.
9	

شکل 5-9: ترکیب توابع LEFT و RIGHT با توابع UPPER و LOWER

فصل 5: کارکردن با توابع متنی

شکل 5-9 نشان می‌دهد که چگونه توابع LEFT و RIGHT با توابع UPPER و LOWER ترکیب می‌شوند تا یک فرمول تولید کنند که متن را به حالت جمله‌ای تبدیل می‌کند. این فرمول به گونه‌ای طراحی شده است که حرف اول جمله را به حروف بزرگ و بقیه متن را به حروف کوچک تبدیل کند. این فرآیند به کاربر کمک می‌کند تا متون را به صورت منظم و استاندارد نمایش دهد.

فرمول تبدیل تاریخ

اگر داده‌های مربوط به سرور یا مین فریم را به کاربرگ‌های خود وارد کنید، یا اگر داده‌هایی از خدمات آنلاین مانند قیمت‌های بازار سهام وارد کنید، اغلب با فرمت‌های تاریخی مواجه می‌شوید که اکسل قادر به شناسایی آن‌ها نیست. یکی از مثال‌های متداول فرمت YYYYMMDD است (به عنوان مثال، 20220823).

برای تبدیل این مقدار به تاریخی که اکسل بتواند با آن کار کند، می‌توانید از توابع LEFT، MID و RIGHT استفاده کنید. اگر تاریخ ناشناخته در سلول A1 قرار داشته باشد، فرمول‌های زیر عمل می‌کنند:

LEFT(A1, 4) سال را استخراج می‌کند.

MID(A1, 5, 2) ماه را استخراج می‌کند.

RIGHT(A1, 2) روز را استخراج می‌کند.

قراردادن این توابع در تابع DATE تاریخ موردقبولی را برای اکسل فراهم می‌آورد:

=DATE(LEFT(A1, 4), MID(A1, 5, 2), RIGHT(A1, 2))

برای یادگیری بیشتر در مورد تابع DATE، به فصل 8، «کار with functions و ساعت» مراجعه کنید.

مطالعه موردی: تولید شماره حساب، بخش اول

بسیاری از شرکت‌ها شماره‌های حساب تأمین‌کننده یا مشتری را با ترکیب بخشی از نام حساب با یک ارزش عددی تولید می‌کنند. توابع متنی اکسل این امکان را برای شما فراهم می‌کنند که به راحتی این شماره‌های حساب را به طور خودکار تولید کنید.

برای شروع، بیایید سه حرف اول نام شرکت را استخراج کرده و آن‌ها را به حروف بزرگ تبدیل کنیم تا آسان‌تر خوانده شوند (فرض بر این است که نام در سلول A2 قرار دارد):

=UPPER(LEFT(A2, 3))

فصل 5: کارکردن با توابع متنی

با این فرمول، می‌توانیم بخشی از نام شرکت را به طور مؤثر به دست آوریم و برای مراحل بعدی استخراج یا پردازش استفاده کنیم.

در مرحله بعد، بخش عددی شماره حساب را با استفاده از شماره ردیف استخراج می‌کنیم:

ROW(A2)

با این حال، برای حفظ طول یکنواخت تمام شماره‌های حساب، بهتر است از تابع TEXT برای پر کردن شماره ردیف با صفرهای اضافی استفاده کنید:

TEXT(ROW(A2), "0000")

شما می‌توانید دو بخش را با هم ترکیب کنید تا شماره حساب کامل به دست آید:

=UPPER(LEFT(A2, 3)) & TEXT(ROW(A2), "0000")

این فرمول سه حرف اول نام شرکت را به حروف بزرگ تبدیل کرده و به آن شماره ردیف که با صفرهای اضافی پر شده است، الحاق می‌کند.

شکل 5-10 مثال‌هایی از این فرمول را نشان می‌دهد که به شما امکان می‌دهد شماره‌های حساب یکنواخت و قابل شناسایی برای تأمین‌کنندگان یا مشتریان خود تولید کنید. این روش به سادگی مدیریت و شناسایی حساب‌ها را تسهیل می‌کند.

B2 fx =UPPER(LEFT(A2, 3)) & TEXT(ROW(A2), "0000")				
	A	B	C	D
1	Company Name	Account Number		
2	Exotic Liquids	EXO0002		
3	New Orleans Cajun Delights	NEW0003		
4	Grandma Kelly's Homestead	GRA0004		
5	Tokyo Traders	TOK0005		
6	Cooperativa de Quesos 'Las Cabras'	COO0006		
7	Mayumi's	MAY0007		
8	Pavlova, Ltd.	PAV0008		
9	Specialty Biscuits, Ltd.	SPE0009		
10	Bigfoot Breweries	BIG0010		

شکل 5-10: تولید خودکار شماره حساب

فصل 5: کارکردن با توابع متنی

این کاربرد از توابع LEFT، UPPER و TEXT استفاده می کند تا به طور خودکار شماره های حساب را از نام های شرکت ها تولید کند. با ترکیب این توابع، نام شرکت ها به حروف بزرگ تبدیل شده و به دنبال آن بخش عددی شماره حساب که با صفرهای اضافی پر شده، اضافه می شود. این فرایند تولید شماره های حساب یکنواخت و قابل شناسایی را ساده و مؤثر می سازد.

جستجو برای زیررشته ها

شما می توانید توابع متنی اکسل را با جستجوی زیررشته ها در یک متن خاص تقویت کنید. به عنوان مثال، در یک رشته که شامل نام و نام خانوادگی یک شخص است، می توانید موقعیت فاصله بین نام ها را پیدا کنید و از این واقعیت برای استخراج نام یا نام خانوادگی استفاده کنید.

توابع FIND و SEARCH

جستجو برای زیررشته ها با استفاده از توابع FIND و SEARCH انجام می شود:

:FIND

FIND(find_text, within_text [, start_num])

:SEARCH

SEARCH(find_text, within_text [, start_num])

پارامترها:

find_text: زیررشته ای که می خواهید جستجو کنید.

within_text: رشته ای که در آن می خواهید جستجو کنید.

start_num: موقعیت کاراکتر که می خواهید جستجو را از آن شروع کنید (پیش فرض 1 است).

تفاوت بین FIND و SEARCH

تابع FIND جستجو را به حروف بزرگ و کوچک حساس می کند، در حالی که SEARCH حساس به حروف بزرگ و کوچک نیست و قابلیت استفاده از کاراکترهای جایگزین (wildcards) را نیز دارد.

با استفاده از این توابع می توانید موقعیت زیررشته ها را در متن پیدا کرده و با آن ها عملیات مختلفی انجام دهید.

فصل 5: کارکردن با توابع متنی

نکات مهم در استفاده از توابع FIND و SEARCH

هنگام استفاده از توابع FIND و SEARCH، به نکات زیر توجه کنید:

- بازگشت موقعیت کاراکتر: این توابع موقعیت کاراکتر اولین بروز (پس از موقعیت کاراکتر start_num) از find_text را در within_text برمی گردانند.
- جستجوی غیرحساس به حروف بزرگ و کوچک: از تابع SEARCH برای جستجوهای غیرحساس به حروف بزرگ و کوچک استفاده کنید. به عنوان مثال: SEARCH("e", "Expenses") ; برمی گرداند 1
- جستجوی حساس به حروف بزرگ و کوچک: از تابع FIND برای جستجوهای حساس به حروف بزرگ و کوچک استفاده کنید. به عنوان مثال: FIND("e", "Expenses") ; برمی گرداند 4
- خطای #VALUE!: اگر find_text در within_text وجود نداشته باشد، این توابع خطای #VALUE! را برمی گردانند.
- استفاده از کاراکترهای جایگزین (wildcards): در find_text آرگومان تابع SEARCH، می توانید از یک علامت سؤال (?) برای مطابقت با هر کاراکتر واحد استفاده کنید.
- در find_text آرگومان تابع SEARCH، می توانید از یک ستاره (*) برای مطابقت با هر تعداد کاراکتر استفاده کنید.
- شامل کردن کاراکترهای خاص: برای گنجاندن کاراکترهای؟ یا * در عملیات SEARCH، هر مورد را در آرگومان find_text با یک علامت تیلدا (~) پیشی ببرید. اگر می خواهید یک کاراکتر تیلدا جستجو کنید، از دو تیلدا (~~) استفاده کنید.

این نکات به شما کمک می کند تا از توابع FIND و SEARCH به طور مؤثرتر و با درک بهتری از قابلیت ها و محدودیت های آنها استفاده کنید.

استخراج نام اول یا نام خانوادگی

اگر شما دارای یک محدوده از سلول ها باشید که شامل نام های اول و نام های خانوادگی افراد است، استخراج این نام ها از هر رشته می تواند بسیار مفید باشد. به عنوان مثال، ممکن است بخواهید نام های اول و نام های خانوادگی را در محدوده های جداگانه ذخیره کنید تا بعداً آنها را به یک جدول پایگاه داده وارد کنید. یا شاید نیاز داشته باشید یک محدوده جدید با ساختار "نام خانوادگی، نام اول" ایجاد کنید تا نام ها را مرتب کنید.

فصل 5: کار کردن با توابع متنی

استفاده از توابع SEARCH و LEFT/RIGHT

راه حل این است که از تابع SEARCH (یا FIND) برای پیدا کردن فاصله ای که نام اول و نام خانوادگی را جدا می کند، استفاده کنید و سپس از تابع LEFT برای استخراج نام اول یا از تابع RIGHT برای استخراج نام خانوادگی استفاده کنید. (البته به یاد داشته باشید که اکسل ویژگی پر کردن خودکار (Flash Fill) را ارائه می دهد که معمولاً برای کارهایی مانند استخراج نامها ساده تر است).

استخراج نام اول

برای استخراج نام اول، می توانید از فرمول زیر استفاده کنید (فرض بر این است که نام کامل در سلول A2 قرار دارد):

=LEFT(A2, SEARCH(" ", A2) - 1)

توجه داشته باشید که این فرمول 1 را از نتیجه SEARCH(" ", A2) کم می کند تا فضای موجود در زیررشته استخراج شده را شامل نشود. شما می توانید از این فرمول در شرایط عمومی تری برای استخراج اولین کلمه از هر رشته چندکلمه ای استفاده کنید.

استخراج نام خانوادگی

برای استخراج نام خانوادگی، نیاز به ساخت فرمول مشابهی با استفاده از تابع RIGHT دارید:

=RIGHT(A2, LEN(A2) - SEARCH(" ", A2))

برای استخراج تعداد صحیح حروف، این فرمول طول رشته اصلی را محاسبه کرده و موقعیت فاصله را از آن کسر می کند. شما می توانید از این فرمول در شرایط عمومی تری برای استخراج دومین کلمه در هر رشته دوکلمه ای استفاده کنید.

شکل 5-11 کاربرگی را نشان می دهد که هر دو فرمول را به کار می گیرد. این فرمولها به شما امکان می دهند تا نامهای اول و نامهای خانوادگی را به راحتی استخراج و مدیریت کنید.

فصل 5: کار کردن با توابع متنی

D2 =TEXTJOIN(" ", TRUE, RIGHT(A2, LEN(A2) - SEARCH(" ", A2)), LEFT(A2, SEARCH(" ", A2) - 1))								
	A	B	C	D	E	F	G	H
1	Full Name	First Name	Last Name	Last Name, First Name				
2	Charlotte Cooper	Charlotte	Cooper	Cooper, Charlotte				
3	Shelley Burke	Shelley	Burke	Burke, Shelley				
4	Regina Murphy	Regina	Murphy	Murphy, Regina				
5	Yoshi Nagase	Yoshi	Nagase	Nagase, Yoshi				
6	Mayumi Ohno	Mayumi	Ohno	Ohno, Mayumi				
7	Ian Devling	Ian	Devling	Devling, Ian				
8	Peter Wilson	Peter	Wilson	Wilson, Peter				
9	Lars Peterson	Lars	Peterson	Peterson, Lars				
10	Carlos Diaz	Carlos	Diaz	Diaz, Carlos				
11	Petra Winkler	Petra	Winkler	Winkler, Petra				
12	Martin Bein	Martin	Bein	Bein, Martin				

شکل 5-11: استخراج نام اول و نام خانوادگی

در این شکل، از توابع LEFT و SEARCH برای استخراج نام اول استفاده شده است؛ همچنین از توابع RIGHT و SEARCH برای استخراج نام خانوادگی بهره گرفته شده است. این فرمول‌ها به کاربر این امکان را می‌دهند که به طور مؤثر نام‌های افراد را از یک‌رشته کامل جداسازی کند و آن‌ها را در محدوده‌های جداگانه قرار دهد. این فرایند مدیریت داده‌ها و استفاده از آن‌ها در برنامه‌های مختلف، مانند پایگاه‌های داده، را ساده‌تر می‌کند.

⚠ هشدار: این فرمول‌ها در هر رشته‌ای که فقط یک کلمه داشته باشد، خطا ایجاد می‌کنند. برای

جلوگیری از این مشکل، می‌توانید از تابع IFERROR استفاده کنید:

=IFERROR(LEFT(A2, FIND(" ", A2) - 1), A2)

توضیحات: با استفاده از این فرمول، اگر سلول حاوی یک فاصله باشد، همه چیز به خوبی پیش می‌رود و فرمول به طور عادی اجرا می‌شود. اگر سلول فاقد فاصله باشد، تابع FIND یک خطا را برمی‌گرداند؛ بنابراین، تابع IFERROR به جای بازگشت نتیجه فرمول، متن موجود در سلول را باز می‌گرداند. این روش باعث می‌شود که فرمول شما در مواقعی که نامی فقط یک کلمه دارد، به درستی عمل کند و از خطا جلوگیری شود.

استخراج نام اول، نام خانوادگی و حرف اول نام میانی

اگر نام کاملی که با آن کار می‌کنید شامل حرف اول نام میانی شخص باشد، فرمول استخراج نام اول همانند قبل خواهد ماند. اما برای پیدا کردن نام خانوادگی باید فرمول را تنظیم کنید. یکی از روش‌های انجام این کار استفاده از ترفندهای مفید FIND و SEARCH است.

فصل 5: کارکردن با توابع متنی

مثال: فرض کنید رشته‌ای که با آن کار می‌کنید به شکل زیر باشد:

Karen E. McHammond

اگر فرض کنیم این رشته در سلول A2 قرار دارد، فرمول زیر:

=SEARCH(" ", A2)

موقعیت فضای اول (6) را برمی‌گرداند. برای پیدا کردن موقعیت فضای دوم، می‌توانید به جای استفاده از موقعیت پیشفرض، آرگومان start_num در تابع SEARCH را به 7 تنظیم کنید—یا به‌طور کلی به موقعیت فضای اول به علاوه 1:

=SEARCH(" ", A2, FIND(" ", A2) + 1)

استخراج نام خانوادگی

حالا می‌توانید این نتیجه را در تابع RIGHT استفاده کنید تا نام خانوادگی را استخراج کنید:

=RIGHT(A2, LEN(A2) - SEARCH(" ", A2, SEARCH(" ", A2) + 1))

توضیحات:

SEARCH(" ", A2) برای پیدا کردن موقعیت فضای اول استفاده می‌شود.

SEARCH(" ", A2, SEARCH(" ", A2) + 1) برای پیدا کردن موقعیت فضای دوم استفاده می‌شود.

نتیجه به تابع RIGHT داده می‌شود تا نام خانوادگی را بر اساس طول کل رشته و موقعیت فضای دوم استخراج کند.

این روش به شما این امکان را می‌دهد که به راحتی نام خانوادگی را حتی در حالتی که نام میانی وجود دارد، استخراج کنید.

استخراج حرف اول نام میانی

برای استخراج حرف اول نام میانی، می‌توانید به جستجوی نقطه (.) بپردازید و از تابع MID برای استخراج حرف قبل از آن استفاده کنید:

=MID(A2, SEARCH(".", A2) - 1, 1)

توضیحات فرمول:

SEARCH(".", A2): موقعیت نقطه را در رشته پیدا می‌کند.

فصل 5: کارکردن با توابع متنی

1 - SEARCH(" ", A2): یک موقعیت کاراکتر عقب‌تر به ما می‌دهد که به ما اجازه می‌دهد حرف اول نام میانی را بگیریم.

1) MID(A2,... , 1): یک کاراکتر را از رشته A2 بر اساس موقعیت محاسبه شده استخراج می‌کند.

شکل 5-12 کاربرگی را نشان می‌دهد که این تکنیک‌ها را به نحوی مؤثر به نمایش می‌گذارد. این فرمول‌ها به شما این امکان را می‌دهند که به راحتی نام اول، نام خانوادگی و حرف اول نام میانی را از یک رشته شامل نام کامل استخراج کنید. این فرایند می‌تواند به طور مؤثری در مدیریت و پردازش داده‌ها کمک کند.

D2 X ✓ fx =RIGHT(A2, LEN(A2) - SEARCH(" ", A2, SEARCH(" ", A2) + 1))				
	A	B	C	D
1	Full Name	First Name	Middle Initial	Last Name
2	Charlotte P. Cooper	Charlotte	P	Cooper
3	Shelley R. Burke	Shelley	R	Burke
4	Regina O. Murphy	Regina	O	Murphy
5	Yoshi H. Nagase	Yoshi	H	Nagase
6	Mayumi U. Ohno	Mayumi	U	Ohno
7	Ian F. Devling	Ian	F	Devling
8	Peter W. Wilson	Peter	W	Wilson
9	Lars X. Peterson	Lars	X	Peterson
10	Carlos B. Diaz	Carlos	B	Diaz
11	Petra Q. Winkler	Petra	Q	Winkler

شکل 5-12: جستجو بعد از اولین وقوع زیررشته

در شکل 5-12، نشان داده می‌شود که چگونه می‌توان از تابع SEARCH پس از اولین وقوع یک زیررشته استفاده کرد تا دومین وقوع آن زیررشته را پیدا کنیم. این روش به ویژه در استخراج اطلاعات از نام‌های کامل، از جمله نام اول، نام خانوادگی و حرف اول نام میانی، مؤثر است. با استفاده از این تکنیک، می‌توان به راحتی فضای دوم یا هر زیررشته دیگری را که بعد از اولین وقوع آن قرار دارد، شناسایی کرد. این فرایند به شما این امکان را می‌دهد که اطلاعات را به شکل سازمان‌یافته و کارآمدتری مدیریت کنید، به ویژه در داده‌های ورودی نام‌ها که شامل نام‌های چندکلمه‌ای هستند. این رویکرد به کاربر کمک می‌کند تا به‌طور مؤثری داده‌ها را استخراج و پردازش کند، به‌خصوص در مواقعی که با نام‌های پیچیده‌تر سروکار دارید.

تعیین حرف ستون

تابع COLUMN در اکسل شماره ستون یک سلول مشخص را برمی‌گرداند. به‌عنوان مثال، برای یک سلول در ستون A، COLUMN عدد 1 را برمی‌گرداند. این موضوع می‌تواند مفید باشد، اما در برخی موارد، ممکن است بخواهید حرف واقعی ستون را بدانید.

فصل 5: کار کردن با توابع متنی

استخراج حرف ستون

این کار کمی پیچیده است؛ زیرا حروف ستون از A تا Z، سپس از AA تا AZ و غیره ادامه دارد. با این حال، تابع CELL در اکسل می تواند (در میان چیزهای دیگر) آدرس یک سلول مشخص را در فرمت مطلق برگرداند - مثلاً A\$2\$ یا AB\$10\$.

مراحل استخراج حرف ستون

برای استخراج حرف ستون، باید زیررشته ای را که بین دو علامت دلار (\$) است، استخراج کنیم. به طور واضح، این زیررشته همیشه از موقعیت دوم شروع می شود. می توان با فرمول زیر شروع کرد:

=MID(CELL("Address", A2), 2, num_chars)

num_chars مقدار آن بستگی به ستونی دارد که در آن هستید و می تواند 1، 2 یا 3 باشد.

تعیین تعداد کاراکترها (num_chars)

موقعیت علامت دلار دوم می تواند بسته به ستون 3، 4 یا 5 باشد. به عبارت دیگر، طول زیررشته همیشه دو واحد کمتر از موقعیت علامت دلار دوم خواهد بود. بنابراین، عبارت زیر مقدار num_chars را فراهم می آورد:

SEARCH("\$", CELL("Address", A2), 3) - 2

حال، فرمول نهایی برای استخراج حرف ستون به شکل زیر خواهد بود:

=MID(CELL("Address", A2), 2, SEARCH("\$", CELL("Address", A2), 3) - 2)

فرمول برای دریافت حرف ستون سلول جاری

اگر بخواهید حرف ستون سلول جاری را بگیرید، می توانید از فرمول کوتاه تری استفاده کنید:

=MID(CELL("Address"), 2, SEARCH("\$", CELL("Address"), 3) - 2)

نکات مهم: این فرمول ها شما را قادر می سازند تا به راحتی حرف ستون را از آدرس سلول استخراج کنید که می تواند در موارد مختلف تجزیه و تحلیل داده کاربردی باشد. استفاده از توابع MID و SEARCH در ترکیب با CELL می تواند به شما کمک کند تا اطلاعات بیشتری در مورد موقعیت های داده های خود به دست آورید.

فصل 5: کارکردن با توابع متنی

تابع REPLACE در اکسل

برنامه‌های Office (و به‌ویژه اکثر برنامه‌های ویندوز) دارای دستوری به نام Replace هستند که به شما اجازه می‌دهد متنی را جستجو کرده و آن را با رشته دیگری جایگزین کنید. مجموعه توابع کاربرگ اکسل نیز دارای چنین ویژگی است که در قالب توابع REPLACE و SUBSTITUTE ارائه می‌شود.

نحو تابع REPLACE

ساختار تابع REPLACE به شکل زیر است:

REPLACE(old_text, start_num, num_chars, new_text)

پارامترها:

old_text: رشته اصلی که شامل زیررشته‌ای است که می‌خواهید آن را جایگزین کنید.

start_num: موقعیت کاراکتر که می‌خواهید از آنجا جایگزینی را شروع کنید.

num_chars: تعداد کاراکترهایی که می‌خواهید جایگزین کنید.

new_text: زیررشته‌ای که می‌خواهید به‌عنوان جایگزین استفاده کنید.

چالش‌های start_num و num_chars

بخشی از این تابع که ممکن است دشوار باشد، تعیین مقادیر start_num و num_chars است. برای مثال، فرض کنید رشته زیر را داریم:

Expense Budget for 2021

برای جایگزین کردن 2021 با 2022 و فرض بر اینکه رشته در سلول A1 قرار دارد، فرمول زیر این کار را انجام می‌دهد:

=REPLACE(A1, 20, 4, "2022")

محاسبه start_num و num_chars

محاسبه مقادیر start_num و num_chars به‌صورت دستی کاری دشوار است و در شرایط عمومی ممکن است حتی این مقادیر را ندانید؛ بنابراین، نیاز به محاسبه آن‌ها دارید:

فصل 5: کارکردن با توابع متنی

برای تعیین مقدار start_num: از تابع FIND یا SEARCH برای پیدا کردن زیررشته‌ای که می‌خواهید جایگزین کنید، استفاده کنید.

برای تعیین مقدار num_chars: از تابع LEN برای به دست آوردن طول متن جایگزین استفاده کنید.

فرمول اصلاح شده

فرمول اصلاح شده به این شکل خواهد بود (فرض بر این است که رشته اصلی در A1 و رشته جایگزین در A2 قرار دارد):

=REPLACE(A1, FIND("2021", A1), LEN("2021"), A2)

توضیحات فرمول

FIND("2021", A1): موقعیت ابتدای زیررشته‌ای که می‌خواهید جایگزین کنید (در اینجا 2021) را برمی‌گرداند.

LEN("2021"): طول زیررشته‌ای که می‌خواهید جایگزین کنید را محاسبه می‌کند.

A2: رشته جدیدی که به جای 2021 قرار می‌گیرد.

این روش قابل انعطاف باعث می‌شود که جایگزینی متن در اکسل به سادگی و بدون نیاز به محاسبه دستی مقادیر کلیدی انجام شود.

تابع SUBSTITUTE در اکسل

به دلیل مراحل اضافی که در تابع REPLACE وجود دارد، این تابع ممکن است کمی پیچیده به نظر برسد. به همین دلیل، بسیاری از کاربران از تابع SUBSTITUTE که ساده‌تر و کاربرپسندتر است، استفاده می‌کنند.

ساختار تابع SUBSTITUTE به شکل زیر است:

SUBSTITUTE(text, old_text, new_text [, instance_num])

پارامترها:

text: رشته اصلی که شامل زیررشته‌ای است که می‌خواهید آن را جایگزین کنید.

old_text: زیررشته‌ای که می‌خواهید جایگزین کنید.

new_text: زیررشته‌ای که می‌خواهید به عنوان جایگزین استفاده کنید.

فصل 5: کارکردن با توابع متنی

instance_num: وقوع خاصی که می‌خواهید جایگزین شود. (پیش‌فرض برای جایگزینی همه وقوع‌ها است).

مثال استفاده از SUBSTITUTE

فرمول زیر همان کاری را انجام می‌دهد که در مثال قبلی با استفاده از تابع REPLACE انجام شد:

=SUBSTITUTE(A1, "2021", "2022")

این فرمول در سلول A1 مقدار 2021 را با 2022 جایگزین می‌کند.

حذف یک کاراکتر از رشته

شما می‌توانید از تابع SUBSTITUTE برای حذف تمام وقوع‌های یک کاراکتر خاص از یک رشته نیز استفاده کنید. به‌عنوان مثال، ممکن است بخواهید تمام فضاها را از یک رشته یا یک آپاستروف را از یک نام حذف کنید.

فرمول عمومی برای حذف کاراکتر به این شکل است:

=SUBSTITUTE(text, character, "")

در اینجا، به جای text رشته اصلی و به جای character کاراکتری که می‌خواهید حذف کنید، قرار می‌دهید.

برای مثال، برای حذف تمام فضاها از رشته موجود در سلول A1، می‌توانید از فرمول زیر استفاده کنید:

=SUBSTITUTE(A1, " ", "")

این فرمول تمامی فضاها را از رشته موجود در A1 حذف می‌کند و نتیجه‌ای بدون فاصله به شما می‌دهد.

نکات مهم: تابع SUBSTITUTE به شما این امکان را می‌دهد که به‌راحتی زیررشته‌ها را جایگزین کنید یا کاراکترهای خاصی را از یک رشته حذف کنید. استفاده از SUBSTITUTE ساده‌تر از REPLACE بوده و برای بسیاری از سناریوهای پردازش متنی مفید است.

فصل 5: کارکردن با توابع متنی

✦ یادداشت: یک کاربرد جالب از تابع SUBSTITUTE این است که می توان از آن برای شمارش تعداد کاراکترهایی که در یک رشته ظاهر می شوند، استفاده کرد. ترفند اینجا این است که اگر یک کاراکتر خاص را از یک رشته حذف کنید، تفاوت در طول بین رشته اصلی و رشته ای که کاراکتر از آن حذف شده، برابر با تعداد دفعاتی است که آن کاراکتر در رشته اصلی وجود داشته است. به عنوان مثال، رشته "expenses" دارای هشت کاراکتر است. اگر تمام "e" ها را حذف کنید، رشته حاصل "xpns" خواهد بود که دارای پنج کاراکتر است. تفاوت 3 است که نشان دهنده تعداد "e" ها در رشته اصلی است. برای محاسبه این در یک فرمول، از تابع LEN استفاده کرده و طول رشته ای که کاراکتر حذف شده از آن گرفته شده را از طول رشته اصلی کسر کنید. فرمولی که تعداد "e" ها را برای رشته ای در سلول A1 شمارش می کند به شکل زیر است:

$$=LEN(A1) - LEN(SUBSTITUTE(A1, "e", ""))$$

توضیحات فرمول:

LEN(A1): طول رشته اصلی را محاسبه می کند (در این مثال یعنی طول "expenses" که برابر با 8 است).
SUBSTITUTE(A1, "e", ""): تمام "e" ها را از رشته حذف می کند، بنابراین رشته حاصل "xpns" خواهد بود.

LEN(SUBSTITUTE(A1, "e", "")): طول رشته بدون "e" ها را محاسبه می کند (که برابر با 5 است).

تفاوت LEN: با کسر طول رشته بدون "e" ها از طول رشته اصلی، تعداد "e" ها به دست می آید:

$$(3 = 5 - 8)$$

این روش برای هر کاراکتری که می خواهید شمارش کنید، کار می کند. فقط کافیست کاراکتر موردنظر را در فرمول تغییر دهید. استفاده از این روش می تواند در تجزیه و تحلیل داده ها و بررسی تکرار کاراکترها در متن های مختلف بسیار مفید باشد.

فصل 5: کار کردن با توابع متنی

حذف دو کاراکتر مختلف از یک رشته

حذف دو کاراکتر مختلف از یک رشته امکان پذیر است و می توان یک تابع SUBSTITUTE را درون دیگری قرار داد تا دو کاراکتر مختلف را از یک رشته حذف کند. به عنوان مثال، ابتدا به عبارت زیر توجه کنید که از SUBSTITUTE برای حذف نقطه ها از یک رشته استفاده می کند:

SUBSTITUTE(A1, ".", "")

از آنجاکه این عبارت یک رشته را برمی گرداند، می توانید نتیجه آن را به عنوان آرگومان text در یک تابع SUBSTITUTE دیگر استفاده کنید؛ بنابراین، فرمول زیر نقطه ها و فضاها را از رشته ای در سلول A1 حذف می کند:

=SUBSTITUTE(SUBSTITUTE(A1, ".", ""), " ", "")

مطالعه موردی: تولید شماره حساب، قسمت II

فرمولی که قبلاً به شما نشان دادم برای تولید خودکار شماره حساب از نام حساب، تنها در صورتی شماره های معتبر تولید می کند که سه حرف اول نام، حروف الفبا باشند. اگر اسامی ای داشته باشید که شامل کاراکترهای دیگری به جز حروف باشند، باید قبل از تولید شماره حساب، آن کاراکترها را حذف کنید. برای مثال، اگر نام حسابی مانند J. D. BigBelly دارید، باید نقطه ها و فضاها را قبل از تولید شماره حساب حذف کنید. می توانید این کار را با افزودن عبارت موجود در بخش قبلی به فرمول تولید شماره حساب از فصل گذشته انجام دهید. به طور خاص، شما آدرس سلول را در تابع LEFT با توابع SUBSTITUTE تودرتو جایگزین می کنید. توجه کنید که این فرمول همچنان بر روی نام های حسابی که با سه حرف شروع می شوند کار می کند.

B2		=UPPER(LEFT(SUBSTITUTE(SUBSTITUTE(A2, ".", ""), " ", ""), 3)) & TEXT(ROW(A2), "0000")							
	A	B	C	D	E	F	G	H	I
1	Company Name	Account Number							
2	J. D. BigBelly	JDB0002							
3	PB Knäckebröd AB	PBK0003							
4	A. Axelrod & Associates	AAX0004							
5	Bigfoot Breweries	BIG0005							
6									

شکل ۵-۱۳ این worksheet از توابع SUBSTITUTE تودرتو برای حذف نقطه ها و فضاها از نام های حساب پیش از تولید شماره های حساب استفاده می کند.

فصل 5: کارکردن با توابع متنی

حذف خط‌های جدید

در بخش‌های قبلی این فصل، با تابع CLEAN آشنا شدید که کاراکترهای غیرقابل چاپ را از یک رشته حذف می‌کند. در مثال، از CLEAN برای حذف خط‌های جدید از یک ورودی چندخطی در یک سلول استفاده کردم. با این حال، ممکن است متوجه یک مشکل کوچک در نتیجه شده باشید: هیچ فضای بین انتهای یک خط و شروع خط بعدی وجود نداشت (به شکل ۴-۵ مراجعه کنید).

اگر تنها نگران خط‌های جدید هستید، به جای تابع CLEAN از فرمول SUBSTITUTE زیر استفاده کنید:

=SUBSTITUTE(A2, CHAR(10), " ")

این فرمول کاراکتر خط جدید (کد ANSI ۱۰) را با یک فضا جایگزین می‌کند و در نتیجه یک رشته مناسب به دست می‌آورد، همان‌طور که در شکل ۴-۵ نشان داده شده است.

B2 X ✓ fx =SUBSTITUTE(A2, CHAR(10), " ")		
	A	B
1	Original String	Line Feeds Removed
	49 Gilbert St. London, England	
2	EC1 4SD	49 Gilbert St. London, England EC1 4SD
	P.O. Box 78934 New Orleans, LA	
3	70117	P.O. Box 78934 New Orleans, LA 70117
	707 Oxford Rd. Ann Arbor, MI	
4	48104	707 Oxford Rd. Ann Arbor, MI 48104
5		

شکل ۴-۵ این worksheet از SUBSTITUTE برای جایگزینی هر کاراکتر خط جدید با یک فضا استفاده می‌کند.