> **Return type**
>> *Table*

property padding: `Tuple[int, int, int, int]`
> Get cell padding.

property row_count: `int`
> Get the current number of rows.

23.31 rich.text

class `rich.text.`**`Text`**(*text='', style='', *, justify=None, overflow=None, no_wrap=None, end='\n',*
tab_size=None, spans=None)

> Text with color / style.
>
>> **Parameters**
>>
>> * **text** (`str`, *optional*) – Default unstyled text. Defaults to "".
>>
>> * **style** (`Union[str, Style]`, *optional*) – Base style for text. Defaults to "".
>>
>> * **justify** (`str`, *optional*) – Justify method: "left", "center", "full", "right". Defaults to None.
>>
>> * **overflow** (`str`, *optional*) – Overflow method: "crop", "fold", "ellipsis". Defaults to None.
>>
>> * **no_wrap** (`bool`, *optional*) – Disable text wrapping, or None for default. Defaults to None.
>>
>> * **end** (`str`, *optional*) – Character to end text with. Defaults to "\n".
>>
>> * **tab_size** (`int`) – Number of spaces per tab, or `None` to use `console.tab_size`. Defaults to None.
>>
>> * **spans** (`List[Span]`, *optional*)

align(*align*, *width*, *character=' '*)

> Align text to a given width.
>
>> **Parameters**
>>
>> * **align** (`AlignMethod`) – One of "left", "center", or "right".
>>
>> * **width** (`int`) – Desired width.
>>
>> * **character** (`str`, *optional*) – Character to pad with. Defaults to " ".
>>
>> **Return type**
>>> None

append(*text*, *style=None*)

> Add text with an optional style.
>
>> **Parameters**
>>
>> * **text** (`Union[Text, str]`) – A str or Text to append.
>>
>> * **style** (`str`, *optional*) – A style name. Defaults to None.
>>
>> **Returns**
>>> Returns self for chaining.