- **style** (*Union[str, Style], optional*) – Base style for text. Defaults to "".
- **emoji** (*bool, optional*) – Also render emoji code. Defaults to True.
- **emoji_variant** (*str, optional*) – Optional emoji variant, either "text" or "emoji". Defaults to None.
- **justify** (*str, optional*) – Justify method: "left", "center", "full", "right". Defaults to None.
- **overflow** (*str, optional*) – Overflow method: "crop", "fold", "ellipsis". Defaults to None.
- **end** (*str, optional*) – Character to end text with. Defaults to "\n".

Returns
A Text instance with markup rendered.

Return type
Text

get_style_at_offset(*console*, *offset*)

Get the style of a character at give offset.

Parameters

- **console** (*~Console*) – Console where text will be rendered.
- **offset** (*int*) – Offset in to text (negative indexing supported)

Returns
A Style instance.

Return type
Style

highlight_regex(*re_highlight*, *style=None*, *, *style_prefix=''*)

Highlight text with a regular expression, where group names are translated to styles.

Parameters

- **re_highlight** (*Union[re.Pattern, str]*) – A regular expression object or string.
- **style** (*Union[GetStyleCallable, StyleType]*) – Optional style to apply to whole match, or a callable which accepts the matched text and returns a style. Defaults to None.
- **style_prefix** (*str, optional*) – Optional prefix to add to style group names.

Returns
Number of regex matches

Return type
int

highlight_words(*words*, *style*, *, *case_sensitive=True*)

Highlight words with a style.

Parameters

- **words** (*Iterable[str]*) – Words to highlight.
- **style** (*Union[str, Style]*) – Style to apply.
- **case_sensitive** (*bool, optional*) – Enable case sensitive matching. Defaults to True.