

classmethod `on(meta=None, **handlers)`

Create a blank style with meta information.

Example

```
style = Style.on(click=self.on_click)
```

Parameters

- **meta** (*Optional*[*Dict*[*str*, *Any*]], *optional*) – An optional dict of meta information.
- ****handlers** (*Any*) – Keyword arguments are translated in to handlers.

Returns

A Style with meta information attached.

Return type

Style

classmethod `parse(style_definition)`

Parse a style definition.

Parameters

style_definition (*str*) – A string containing a style.

Raises

errors.StyleSyntaxError – If the style definition syntax is invalid.

Returns

A Style instance.

Return type

Style

classmethod `pick_first(*values)`

Pick first non-None style.

Parameters

values (*str* | *Style* | *None*)

Return type

str | *Style*

render(*text*=", *", *color_system*=*ColorSystem.TRUECOLOR*, *legacy_windows*=*False*)

Render the ANSI codes for the style.

Parameters

- **text** (*str*, *optional*) – A string to style. Defaults to "".
- **color_system** (*Optional*[*ColorSystem*], *optional*) – Color system to render to. Defaults to *ColorSystem.TRUECOLOR*.
- **legacy_windows** (*bool*)

Returns

A string containing ANSI style codes.

Return type

str