Python f-string cheat sheets

See fstring.help for more examples and for a more detailed discussion of this syntax see this string formatting article.

All numbers

The below examples assume the following variables:

```
>>> number = 4125.6
>>> percent = 0.3738
```

Example Output	Replacement Field	Fill	Width	Grouping	Precision	Type
`'4125.60'`	`{number:.2f}`				`.2`	`f`
`'4,125.60'`	`{number:,.2f}`			`,`	`.2`	`f`
`'04125.60'`	`{number:08.2f}`	`0`	`8`		`.2`	`f`
`' 4125.60'`	`{number: 8.2f}`		`8`		`.2`	`f`
`'4.1e+03'`	`{number:.2g}`				`.2`	`g`
`'4125.6'`	`{number:.8g}`				`.8`	`g`
`'37%'`	`{percent:.0%}`				`.0`	`%`

These format specifications only work on all numbers (both `int` and `float`).

- Type `f` with precision `.n` displays `n` digits after the decimal point.
- Type `g` with precision `.n` displays `n` significant digits in scientific notation. Trailing zeros are not displayed.

Integers

These examples assume the following variable:

```
>>> number = 10
```

Example Output	Replacement Field	Alt	Fill	Width	Grouping	Type
`'0010'`	`{number:04d}`		`0`	`4`		`d`

Example Output	Replacement Field	Alt	Fill	Width	Grouping	Type
`'  10'`	`{number: 4d}`			`4`		`d`
`'10'`	`{number:,}`				`,`	
`'1010'`	`{number:b}`					`b`
`'a'`	`{number:x}`					`x`
`'A'`	`{number:X}`					`X`
`'0xa'`	`{number:#x}`	`#`				`x`
`'000a'`	`{number:04x}`		`0`	`4`		`x`
`'0000_1010'`	`{number:09_b}`		`0`	`9`	`_`	`b`

An empty type is synonymous with `d` for integers.

These format specifications only work on integers (`int`).

Strings

These examples assume the following variable:

```
>>> string = "Python"
```

Example Output	Replacement Field	Fill Char	Align	Width
`'              Python'`	`{string:>20}`		`>`	`20`
`'Python              '`	`{string:<20}`		`<`	`20`
`'       Python       '`	`{string:^20}`		`^`	`20`
`'0000Python'`	`{string:0>10}`	`0`	`>`	`10`

These format specifications work on strings (`str`) and most other types (any type that doesn't specify its own custom format specifications).

All objects

The below modifiers are special syntaxes supported by all object types. Some format specifications are also included to show how to mix and match these syntaxes with the `:` syntax.

Self-Doc	Conversion	Format Spec	Field	Example Output
	`!s`		`{expression!s}`	`'Hi! ✨'`
	`!r`		`{expression!r}`	`"'Hi! ✨'"`
	`!a`		`{expression!a}`	`"'Hi! \\u2728'"`
	`!r`	`<10`	`{expression!r:<10}`	`"'Hi! ✨'  "`
`=`			`{expression=}`	`"name='Trey'"`
`=`			`{expression = }`	`"name = 'Trey'"`
`=`	`!s`		`{expression=!s}`	`"name=Trey"`
`=`		`.2f`	`{expression=:.2f}`	`'len(name)=4.00'`

An empty conversion field is synonymous with `!s`, unless a self-documenting expression is used. When a self-documenting expression is used, an empty conversion field uses `!r`.

See fstring.help for some examples.

And see this article on string formatting for more details.