Python Format Strings

Enjoy this cheat sheet at its fullest within Dash, the macOS documentation browser.

Field Width and Alignment

`'hey {:10}'.format('hello')`	Specify width (Aign left, fill with spaces)
`'{:010}'.format(2)`	Fill with zeroes Output: `0000000002`
`'{:*^30}'.format('text')`	Specify width, align to center Output: `**********text**********`

Member and Element Access

```
'{0}, {1}, {2}'.format(1, 2, 3)
```

Access arguments by ordinal position

Output: `1, 2, 3`

```
'{}, {}, {}'.format(1, 2, 3)
```

Implicit positional arguments (2.7 and above only)

Output: `1, 2, 3`

```
'{value1}, {value2}, {value2}'.format(value1=1, value2=2, value3=3)
```

Access keyword arguments by name

Output: `1, 2, 2`

```
'{[1]}'.format(['first', 'second', 'third'])
```

Access element by index

Output: `second`

```
'{.name}'.format(sys.stdin)
```

Access element attribute

Output: `<stdin>`

```
'{[name]}'.format({'name': 'something'})
```

Access element by key

Output: `something`

Numerical Representation

`'{:x}'.format(100)`	Hexadecimal representation Output: `64`
`'{:X}'.format(3487)`	Hexadecimal representation (uppercase letters) Output: `D9F`
`'{:#x}'.format(100)`	Hexadecimal representation (including the `0x`) Output: `0x64`
`'{:b}'.format(100)`	Binary representation Output: `1100100`
`'{:c}'.format(100)`	Character representation Output: `d`
`'{:d}'.format(100)`	Decimal representation (default) Output: `100`
`'{:,}'.format(1000000)`	With thousands separator Output: `1,000,000`
`'{:o}'.format(100)`	Octal representation Output: `144`
`'{:n}'.format(100)`	Like `d`, but uses locale information for separators Output: `100`
`'{:e}'.format(0.0000000001)`	Exponent notation

	Output: `1.000000e-10`
`'{:E}'.format(0.0000000001)`	Exponent notation (capital 'E') Output: `1.000000E-10`
`'{:f}'.format(3/14.0)`	Fixed point Output: `0.214286`
`'{:g}'.format(3/14.0)`	General format Output: `0.214286`
`'{:%}'.format(0.66)`	Percentage Output: `66.000000%`
`'{:.3}'.format(0.214286)`	Precision Output: `0.214`

Conversions

`'{!r}'.format('string')`	Calling `repr` on arguments Output: `'string'`
`'{!s}'.format(1.53438987)`	Calling `str` on arguments Output: `1.53438987`

You can modify and improve this cheat sheet here