

---

**Prefixed**  
*Release 0.3.2*

**Aug 31, 2020**



---

## Contents

---

<b>1</b>	<b>SI (Decimal) Prefixes</b>	<b>1</b>
<b>2</b>	<b>IEC (Binary) Prefixes</b>	<b>3</b>
<b>3</b>	<b>Format Specification</b>	<b>5</b>
3.1	Prefixed-specific fields . . . . .	5
<b>4</b>	<b>API Reference</b>	<b>7</b>
<b>5</b>	<b>Overview</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



# CHAPTER 1

---

## SI (Decimal) Prefixes

---

Prefix	Name	Base
Y	Yotta	$10^{24}$
Z	Zetta	$10^{21}$
E	Exa	$10^{18}$
P	Peta	$10^{15}$
T	Tera	$10^{12}$
G	Giga	$10^9$
M	Mega	$10^6$
k	Kilo	$10^3$
m	Milli	$10^{-3}$
$\mu$	Micro	$10^{-6}$
n	Nano	$10^{-9}$
p	Pico	$10^{-12}$
f	Femto	$10^{-15}$
a	Atto	$10^{-18}$
z	Zepto	$10^{-21}$
y	Yocto	$10^{-24}$



## CHAPTER 2

---

### IEC (Binary) Prefixes

---

Prefix	Name	Base
Y	Yobi	$2^{80}$
Z	Zebi	$2^{70}$
E	Exbi	$2^{60}$
P	Pedi	$2^{50}$
T	Tebi	$2^{40}$
G	Gibi	$2^{30}$
M	Mebi	$2^{20}$
K	Kibi	$2^{10}$





---

Format Specification

---

```

format_spec ::= [[fill]align][sign][#][0][!][width][grouping_option] [%[-
↪]margin][.precision][type]
fill        ::= <any character>
align       ::= "<" | ">" | "=" | "^"
sign        ::= "+" | "-" | " "
width       ::= digit+
grouping_option ::= "_" | ","
precision  ::= digit+
type        ::= "e" | "E" | "f" | "F" | "g" | "G" | "h" | "j" | "J" | "n" | "%"

```

Prefix-specific fields are defined below. Descriptions of standard fields can be found in the [Format Specification Mini-Language](#) documentation.

## 3.1 Prefixed-specific fields

### 3.1.1 Flags

Flag	Meaning
'!'	Add a single space between number and prefix

### 3.1.2 Margin

By default, a prefix will be used when the magnitude of that prefix is reached. For example, `format(Float(999), '.1h')` will result in `'999.0'` and `format(Float(1000), '.1h')` will result in `'1.0k'`.

Margin specifies the percentage to raise or lower these thresholds.

```
>>> f'{Float(950):%-5.2h}'  
'0.95k'  
  
>>> f'{Float(1000):%5.2h}'  
'1000.00'
```

### 3.1.3 Presentation Types

Type	Meaning
'h'	SI format. Outputs the number with closest divisible SI prefix. (k, M, G, ...)
'j'	IEC Format. Outputs the number with closest divisible IEC prefix. (Ki, Mi, Gi, ...)
'J'	Short IEC Format. Same as 'j' but only a single character. (K, M, G, ...)

**class** `prefixed.Float` (`[x]`)

Subclass of the built-in `float` class

Key differences:

- When a math operation is performed with another real number type (`float`, `int`), the result will be a `prefixed.Float` instance.
- Additional presentation types `'h'`, `'j'`, and `'J'` are supported for f-strings and `format()`.

Type	Meaning
<code>'h'</code>	SI format. Outputs the number with closest divisible SI prefix. (k, M, G, ...)
<code>'j'</code>	IEC Format. Outputs the number with closest divisible IEC prefix. (Ki, Mi, Gi, ...)
<code>'J'</code>	Short IEC Format. Same as <code>'j'</code> but only a single character. (K, M, G, ...)

- When initializing from strings, SI and IEC prefixes are honored

```
>>> Float('2k')
Float(2000.0)

>>> Float('2Ki')
Float(2048.0)
```

- An additional format flag `'!'` is available which adds a space before the prefix

```
>>> f'{Float(3250):!.2h}'
'3.25 k'
```

- An additional field, `margin`, can be specified which lowers or raises the threshold for for each prefix by the given percentage. `Margin` is specified before precision with the syntax `%[-]digit+`.

```
>>> f'{Float(950):.2h}'  
'950.00'  
  
>>> f'{Float(950):%-5.2h}'  
'0.95k'  
  
>>> f'{Float(1000):%5.2h}'  
'1000.00'  
  
>>> f'{Float(1050):%5.2h}'  
'1.05k'
```

`Prefixed` provides an alternative implementation of the built-in `float` which supports formatted output with SI (decimal) and IEC (binary) prefixes.

```
>>> from prefixed import Float
>>> f'{Float(3250):.2h}'
'3.25k'
>>> '{:.2h}s'.format(Float(.00001534))
'15.34µs'
>>> '{:.2j}B'.format(Float(42467328))
'40.50MiB'
>>> f'{Float(2048):.2J}B'
'2.00KB'
```

Because `prefixed.Float` inherits from the built-in `float`, it behaves exactly the same in most cases.

Key differences:

- When a math operation is performed with another real number type (`float`, `int`), the result will be a `prefixed.Float` instance.
- Additional presentation types `'h'`, `'j'`, and `'J'` are supported for f-strings and `format()`.

Type	Meaning
<code>'h'</code>	SI format. Outputs the number with closest divisible SI prefix. (k, M, G, ...)
<code>'j'</code>	IEC Format. Outputs the number with closest divisible IEC prefix. (Ki, Mi, Gi, ...)
<code>'J'</code>	Short IEC Format. Same as <code>'j'</code> but only a single character. (K, M, G, ...)

- When initializing from strings, SI and IEC prefixes are honored

```
>>> Float('2k')
Float(2000.0)

>>> Float('2Ki')
Float(2048.0)
```

- An additional format flag ‘!’ is available which adds a space before the prefix

```
>>> f'{Float(3250):!.2h}'
'3.25 k'
```

- An additional field, margin, can be specified which lowers or raises the threshold for for each prefix by the given percentage. Margin is specified before precision with the syntax %[-]digit+.

```
>>> f'{Float(950):.2h}'
'950.00'

>>> f'{Float(950):%-5.2h}'
'0.95k'

>>> f'{Float(1000):%5.2h}'
'1000.00'

>>> f'{Float(1050):%5.2h}'
'1.05k'
```

**p**

prefixed, 7





**F**

Float (*class in prefixed*), 7

**P**

prefixed (*module*), 7