

### Code Lay-out

Indentation	4 spaces
Max. Length	79 characters
Bracket/etc. closing	<pre>my_list = [     1, 2, 3,     4, 5, 6, ]</pre>
Break before binary operator	<pre>x = a + b     + c</pre>
Imports	<pre>import _ as _ from _ import _, _</pre>
String Quotes (both correct)	<pre>a = "word" b = 'word'</pre>

### Whitespace

Bracket/etc. sides	<pre>spam(ham[1], {eggs: 2})</pre>
Around an assignment	<pre>x = 4 * y variable = 4</pre>
Between a trailing comma and close parenthesis	<pre>foo = (4,)</pre>
Comma/etc. sides	<pre>if x == 4: print(x, y); x, y = y, x</pre>
Colon sides	<pre>arr[3:6], arr[3:6:] arr[:9]</pre>
Operators surround (try in lowest priority)	<pre>x *= 2 + 1 y = (2+5) * (5-1)</pre>
Keyword argument or default value	<pre>def func(val, age=18):     return num(r=real, i=imag)</pre>
Argument annotation with a default value	<pre>def func(sep: AnyStr = None, age=10):     ...</pre>

### Comments

First word capitalized	<pre># This is a comment</pre>
Block	<pre># This is a # block. Yes. # # It is.</pre>
Inline	<pre>y = x + 1 # Linear</pre>
Docstrings	<pre>def calc_age():     """Return the age     Be careful with the end     """</pre>

### Naming Conventions

Variable/Function	lower_case_with_underscores
Class/Exception	CapWords
Non-public methods and instance variables	<code>_var</code>
Constants	UPPER_CASE_WITH_UNDERSCORES
Dunders	<code>__var__</code>

### Structures

Conditional	<pre>if x == 0:     ... elif x == 1:     ... else:     ...</pre>
Ternary	<pre>y = 0 if x == 0 else x == 1</pre>
Match	<pre>match x:     case 0:         ...     case 1:         ...     case _:         ...</pre>
For in	<pre>for i in list:     ...</pre>
While	<pre>while x &gt;= 0:     ...     x -= 1</pre>
Function	<pre>def calc_age(day, month, year=2004):     ...</pre>
List comprehension	<pre>[x**2 for x in list if x != 0]</pre>
Map	<pre>map(lambda x: x**2, list)</pre>
Filter	<pre>filter(lambda x: x &lt; 0, list)</pre>
Reduce	<pre>reduce((lambda x, y: x * y), list)</pre>

