
I'm Dustin
http://github.com/di


promptworks

wat?

Mind-Bending

Edge Cases

(in Python)


wat?

wat #0

wat #0

```
>>> x = ...
>>> x == x
False
```

POSSIBLE!

wat #0

```
>>> x = ...
>>> x == x
False
```

wat #0 - Possible!

```
>>> x = 0*1e309
>>> x == x
False
```

wat #0 - Possible!

```
>>> x = 0*1e309
>>> x == x
False
>>> x
nan
```

wat #0 - Possible!

```
>>> x = 0*1e309
>>> x == x
False
>>> x
nan
>>> 0*float('inf')
nan
```

wat #0 - Possible!

```
>>> x = 0*1e309
>>> x == x
False
>>> x
nan
>>> 0*float('inf')
nan
>>> float('nan')
nan
```


NaN

A brief word on NaN

This is not Python:

```
> Array(16).join("wat" - 1) + " Batman!"
"NaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaN Batman!"
```

https://www.destroyallsoftware.com/talks/wat

A brief word on NaN

```
>>> 0*1e309
nan
>>> float('nan')
nan
>>> from decimal import Decimal; Decimal('nan')
Decimal('NaN')
>>> complex('nan')
(nan+0j)
```

wat #1

wat #1

```
>>> x = ...
>>> a = ...
>>> b = ...
>>> c = ...
>>> max(x) < max(x[a:b:c])
True
```

NOT POSSIBLE.

wat #1 - Not Possible

```
>>> x = ...
>>> a = ...
>>> b = ...
>>> c = ...
>>> max(x) < max(x[a:b:c])
True
```

wat #2

wat #2

```
>>> x = ...
>>> y = ...
>>> min(x, y) == min(y, x)
False
```

POSSIBLE!

wat #2 - Possible!

```
>>> x = ...
>>> y = ...
>>> min(x, y) == min(y, x)
False
```

wat #2 - Possible!

```
>>> x = {0}
>>> y = ...
>>> min(x, y) == min(y, x)
False
```

wat #2 - Possible!

```
>>> x = {0}
>>> y = {1}
>>> min(x, y) == min(y, x)
False
```

wat #2 - Possible!

```
>>> min({0}, {1})
set([0])
```

wat #2 - Possible!

```
>>> min({0}, {1})
set([0])
>>> min({1}, {0})
set([1])
```

wat #2 - Possible!

```
>>> min({0}, {1})
set([0])
>>> min({1}, {0})
set([1])
>>> min({0, 1}, {0})
set([0])
```

wat #2 - Possible!

```
>>> def min(*args):
...
```

wat #2 - Possible!

```
>>> def min(*args):
...     has_item = False
...     min_item = None
...
```

wat #2 - Possible!

```
>>> def min(*args):
...     has_item = False
...     min_item = None
...     for x in args:
...
```

wat #2 - Possible!

```
>>> def min(*args):
...     has_item = False
...     min_item = None
...     for x in args:
...         if not has_item or x < min_item:
...
```

wat #2 - Possible!

```
>>> def min(*args):
...     has_item = False
...     min_item = None
...     for x in args:
...         if not has_item or x < min_item:
...             has_item = True
...             min_item = x
...
```

wat #2 - Possible!

```
>>> def min(*args):
...     has_item = False
...     min_item = None
...     for x in args:
...         if not has_item or x < min_item:
...             has_item = True
...             min_item = x
...     return min_item
...
```

wat #2 - Possible!

```
>>> 0 < 1
True
```

wat #2 - Possible!

```
>>> 0 < 1
True
>>> {0} < {1}
False
```

wat #2 - Possible!

```
>>> 0 < 1
True
>>> {0} < {1}
False
>>> {0} < {0, 1}
True
```

wat #2 - Possible!

```
>>> 0 < 1
True
>>> {0} < {1}
False
>>> {0} < {0, 1}
True
>>> min({0}, {1})
set([0])
```

wat #3

wat #3

```
>>> x = ...
>>> y = ...
>>> any(x) and not any(x + y)
True
```

NOT POSSIBLE.

wat #3 - Not Possible

```
>>> x = ...
>>> y = ...
>>> any(x) and not any(x + y)
True
```

wat #4

wat #4

```
>>> x = ...
>>> y = ...
>>> x.count(y) > len(x)
True
```

POSSIBLE!

wat #4 - Possible!

```
>>> x = ...
>>> y = ...
>>> x.count(y) > len(x)
True
```

wat #4 - Possible!

```
>>> x = 'foobar'
>>> y = ...
>>> x.count(y) > len(x)
True
```

wat #4 - Possible!

```
>>> x = 'foobar'
>>> y = ''
>>> x.count(y) > len(x)
True
```

wat #4 - Possible!

```
>>> x = 'foobar'
>>> y = ''
>>> x.count(y) > len(x)
True
>>> len('foobar')
6
```

wat #4 - Possible!

```
>>> x = 'foobar'
>>> y = ''
>>> x.count(y) > len(x)
True
>>> len('foobar')
6
>>> 'foobar'.count('')
7
```

wat #4 - Possible!

```
>>> def count(s, sub):
...
```

wat #4 - Possible!

```
>>> def count(s, sub):
...     result = 0
...
```

wat #4 - Possible!

```
>>> def count(s, sub):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...
```

wat #4 - Possible!

```
>>> def count(s='foo', sub='foobar'):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...             # range(3 + 1 - 6)
...             # range(-2)
...             # []
```

wat #4 - Possible!

```
>>> def count(s='foobar', sub='foobar'):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...             # range(6 + 1 - 6)
...             # range(1)
...             # [0]
```

wat #4 - Possible!

```
>>> def count(s='foobar', sub='foo'):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...             # range(6 + 1 - 3)
...             # range(4)
...             # [0, 1, 2, 3]
```

wat #4 - Possible!

```
>>> def count(s='foobar', sub=''):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...             # range(6 + 1 - 0)
...             # range(7)
...             # [0, 1, 2, 3, 4, 5, 6]
```

wat #4 - Possible!

```
>>> def count(s, sub):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...         possible_match = s[i:i + len(sub)]
...
```

wat #4 - Possible!

```
>>> def count(s='foobar', sub='foo'):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...         possible_match = s[i:i + len(sub)]
...                          # s[0:0 + 3]
...                          # s[0:3]
...                          # 'foo'
```

wat #4 - Possible!

```
>>> def count(s='foobar', sub=''):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...         possible_match = s[i:i + len(sub)]
...                          # s[0:0 + 0]
...                          # s[0:0]
...                          # ''
```

wat #4 - Possible!

```
>>> def count(s='foobar', sub=''):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...         possible_match = s[i:i + len(sub)]
...                          # s[6:6 + 0]
...                          # s[6:6]
...                          # ''
```

wat #4 - Possible!

```
>>> def count(s, sub):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...         possible_match = s[i:i + len(sub)]
...         if possible_match == sub:
...             result += 1
...     return result
...
```

wat #4 - Possible!

```
>>> def count(s, sub):
...     result = 0
...     for i in range(len(s) + 1 - len(sub)):
...         possible_match = s[i:i + len(sub)]
...         if possible_match == sub:
...             result += 1
...     return result
...
>>> count('foobar', '')
7
```

wat #5

wat #5

```
>>> x = ...
>>> y = ...
>>> z = ...
>>> x * (y * z) == (x * y) * z
False
```

POSSIBLE!

wat #5 - Possible!

```
>>> x = ...
>>> y = ...
>>> z = ...
>>> x * (y * z) == (x * y) * z
False
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = ...
>>> z = ...
>>> x * (y * z) == (x * y) * z
False
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = -1
>>> z = ...
>>> x * (y * z) == (x * y) * z
False
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = -1
>>> z = -1
>>> x * (y * z) == (x * y) * z
False
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = -1
>>> z = -1
>>> x * (y * z) == (x * y) * z
False
>>> x * (y * z) == [0]*(-1*-1)
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = -1
>>> z = -1
>>> x * (y * z) == (x * y) * z
False
>>> x * (y * z) == [0]*(-1*-1) == [0]*1
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = -1
>>> z = -1
>>> x * (y * z) == (x * y) * z
False
>>> x * (y * z) == [0]*(-1*-1) == [0]*1 == [0]
True
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = -1
>>> z = -1
>>> x * (y * z) == (x * y) * z
False
>>> x * (y * z) == [0]*(-1*-1) == [0]*1 == [0]
True
>>> (x * y) * z == ([0]*-1)*-1
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = -1
>>> z = -1
>>> x * (y * z) == (x * y) * z
False
>>> x * (y * z) == [0]*(-1*-1) == [0]*1 == [0]
True
>>> (x * y) * z == ([0]*-1)*-1 == []*-1
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = -1
>>> z = -1
>>> x * (y * z) == (x * y) * z
False
>>> x * (y * z) == [0]*(-1*-1) == [0]*1 == [0]
True
>>> (x * y) * z == ([0]*-1)*-1 == []*-1 == []
True
```

wat #5 - Possible!

```
>>> x = ...
>>> y = ...
>>> z = ...
>>> x * (y * z) == (x * y) * z
False
```

wat #5 - Possible!

```
>>> x = 5e-234
>>> y = 3
>>> z = 9007199254740993
>>> x * (y * z) == (x * y) * z
False
```

wat #5 - Possible!

```
>>> x = 5e-234
>>> y = 3
>>> z = 9007199254740993
>>> x * (y * z) == (x * y) * z
False
>>> x * (y * z)
1.3350443151043 21e-307
```

wat #5 - Possible!

```
>>> x = 5e-234
>>> y = 3
>>> z = 9007199254740993
>>> x * (y * z) == (x * y) * z
False
>>> x * (y * z)
1.3350443151043208e-307
>>> (x * y) * z
1.3350443151043208e-307
```

wat #6

wat #6

```
>>> x = ...
>>> y = ...
>>> x < y and all(a >= b for a, b in zip(x, y))
True
```

POSSIBLE!

wat #6 - Possible!

```
>>> x = ...
>>> y = ...
>>> x < y and all(a >= b for a, b in zip(x, y))
True
```

wat #6 - Possible!

```
>>> x = ''
>>> y = ...
>>> x < y and all(a >= b for a, b in zip(x, y))
True
```

wat #6 - Possible!

```
>>> x = ''
>>> y = 'foobar'
>>> x < y and all(a >= b for a, b in zip(x, y))
True
```

wat #6 - Possible!

```
>>> x = ''
>>> y = 'foobar'
>>> x < y and all(a >= b for a, b in zip(x, y))
True
>>> '' < 'foobar'
True
```

wat #6 - Possible!

```
>>> x = ''
>>> y = 'foobar'
>>> x < y and all(a >= b for a, b in zip(x, y))
True
>>> '' < 'foobar'
True
>>> zip('', 'foobar')
[]
```

wat #6 - Possible!

```
>>> x = ''
>>> y = 'foobar'
>>> x < y and all(a >= b for a, b in zip(x, y))
True
>>> '' < 'foobar'
True
>>> zip('', 'foobar')
[]
>>> all([])
True
```

wat #7

wat #7

```
>>> x = ...
>>> len(set(list(x))) == len(list(set(x)))
False
```

NOT POSSIBLE.

wat #7 - Not Possible

```
>>> x = ...
>>> len(set(list(x))) == len(list(set(x)))
False
```

wat #8

wat #8

```
>>> x = ...
>>> min(x) == min(*x)
False
```

POSSIBLE!

wat #8 - Possible!

```
>>> x = ...
>>> min(x) == min(*x)
False
```

wat #8 - Possible!

```
>>> x = [[0]]
>>> min(x) == min(*x)
False
```

wat #8 - Possible!

```
>>> x = [[0]]
>>> min(x) == min(*x)
False
>>> min([1, 2, 3]) == min(*[1, 2, 3]) == min(1, 2, 3)
True
```

wat #8 - Possible!

```
>>> x = [[0]]
>>> min(x) == min(*x)
False
>>> min([1, 2, 3]) == min(*[1, 2, 3]) == min(1, 2, 3)
True
>>> min(x) == [0]
True
```

wat #8 - Possible!

```
>>> x = [[0]]
>>> min(x) == min(*x)
False
>>> min([1, 2, 3]) == min(*[1, 2, 3]) == min(1, 2, 3)
True
>>> min(x) == [0]
True
>>> min(*x) == min([0]) == 0
True
```

wat #9

wat #9

```
>>> x = ...
>>> y = ...
>>> sum(0 * x, y) == y
False
```

NOT POSSIBLE.

wat #9 - Not Possible

```
>>> x = ...
>>> y = ...
>>> sum(0 * x, y) == y
False
```

wat #9 - Not Possible

```
>>> x = ...
>>> y = ...
>>> sum(0 * x, y) == y
False
>>> sum([1, 1, 1], 7)
10
```

wat #9 - Not Possible

```
>>> x = ...
>>> y = ...
>>> sum(0 * x, y) == y
False
>>> sum([1, 1, 1], 7)
10
>>> sum([], 7)
7
```

wat #10

wat #10

```
>>> x = ...
>>> y = ...
>>> y > max(x) and y in x
True
```

POSSIBLE!

wat #10 - Possible!

```
>>> x = ...
>>> y = ...
>>> y > max(x) and y in x
True
```

wat #10 - Possible!

```
>>> x = 'aa'
>>> y = ...
>>> y > max(x) and y in x
True
```

wat #10 - Possible!

```
>>> x = 'aa'
>>> y = 'aa'
>>> y > max(x) and y in x
True
```

wat #10 - Possible!

```
>>> x = 'aa'
>>> y = 'aa'
>>> y > max(x) and y in x
True
>>> max('aa')
'a'
```

wat #10 - Possible!

```
>>> x = 'aa'
>>> y = 'aa'
>>> y > max(x) and y in x
True
>>> max('aa')
'a'
>>> 'aa' > 'a'
True
```

wat #10 - Possible!

```
>>> x = 'aa'
>>> y = 'aa'
>>> y > max(x) and y in x
True
>>> max('aa')
'a'
>>> 'aa' > 'a'
True
>>> 'aa' in 'aa'
True
```

wat #10 - Possible!

```
>>> x = 'aa'
>>> y = 'aa'
>>> y > max(x) and y in x
True
>>> max('aa')
'a'
>>> 'aa' > 'a'
True
>>> 'aa' in ['a', 'a']
False
```

One last thing...

Thanks!

https://github.com/di/talks/pygotham_2016/

Questions?

wat #0 - Possible!

```
>>> x = 0*1e309
>>> x == x
False
```

wat #1 - Not Possible

```
>>> x = ...
>>> a = ...
>>> b = ...
>>> c = ...
>>> max(x) < max(x[a:b:c])
True
```

wat #2 - Possible!

```
>>> x = {0}
>>> y = {1}
>>> min(x, y) == min(y, x)
False
```

wat #3 - Not Possible

```
>>> x = ...
>>> y = ...
>>> any(x) and not any(x + y)
True
```

wat #4 - Possible!

```
>>> x = 'foobar'
>>> y = ''
>>> x.count(y) > len(x)
True
```

wat #5 - Possible!

```
>>> x = [0]
>>> y = -1
>>> z = -1
>>> x * (y * z) == (x * y) * z
False
```

wat #5

```
>>> x = 5e-234
>>> y = 3
>>> z = 9007199254740993
>>> x * (y * z) == (x * y) * z
False
```

wat #6 - Possible!

```
>>> x = ''
>>> y = 'foobar'
>>> x < y and all(a >= b for a, b in zip(x, y))
True
```

wat #7 - Not Possible

```
>>> x = ...
>>> len(set(list(x))) == len(list(set(x)))
False
```

wat #8 - Possible!

```
>>> x = [[0]]
>>> min(x) == min(*x)
False
```

wat #9 - Not Possible

```
>>> x = ...
>>> y = ...
>>> sum(0 * x, y) == y
False
```

wat #10 - Possible!

```
>>> x = 'aa'
>>> y = 'aa'
>>> y > max(x) and y in x
True
```