

```
>>> 5 < 6
True
>>> 5 < 5
False
>>> 5 <= 5
True
>>> x = 1000
>>> y = 89
>>> x > y
True
>>> x == y
False
>>> 1000 == x
True
>>> s = "banana"
>>> s == "banana"
True
>>> True
True
>>> False
False
>>> "1" == 1

>>> "a" < "b"
```

## New: comparison operators

< <= > >=

Compare numbers, strings and more. Produces True or False depending on the relationship of the values

==

Takes any two values, produces True if the values are equal

```
>>> True and False
False
>>> True or False
True
>>> True and True
True
>>> False or False
False
>>> False and False
False
>>> False and True
```

```
>>> False or True
```

```
>>> True or True
```

and or

Take two **booleans** and produce a new boolean:

b1 and b2 is True when: \_\_\_\_\_

b1 or b2 is True when: \_\_\_\_\_

```
= RESTART ...
>>> is_positive(9)
True
>>> is_positive(-1)
False
```

```
def is_positive(n):
    return n > 0
```

# write `is_longer_than`, which takes a string `s` and  
# a number `n` and returns True if the string has  
# more than `n` characters, and False otherwise.

# write a function `between`, which takes three  
# numbers `x`, `y`, `z` and returns true if `y` is  
# between `x` and `z`.

```

== RESTART ...
>>> my_abs(-1)
1
>>> my_abs(0)
0
>>> letter_grade1(50)
"F"
>>> letter_grade1(75)
"C"

```

### New: if statements

```

if condition1: body1
elif condition2: body2
elif condition3: body3
...
else: body_else

```

any number of elif clauses (including 0)

optional else clause

To evaluate an if statement, Python follows these steps:

- Evaluate condition1.
  - If True, evaluate body1 (and no further clauses!)
  - If False, go on to the next step
- Evaluate condition2
  - If True, evaluate body2 (and no further clauses!)
  - If False, go onto the next step
- Evaluate condition3
  - If True, evaluate body3 (and no further clauses!)
  - If False, go onto the next step
- ... repeat for all elif clauses ...
- If there's an else clause, evaluate body\_else

### Example trace:

```
letter_grade1(75)
```

```

if points >= 90: return "A"
elif points >= 80: return "B"
elif points >= 70: return "C"
else: return "F"

```

points : 75

```

if False: return "A"
elif points >= 80: return "B"
elif points >= 70: return "C"
else: return "F"

```

points : 75

```

if False: return "A"
elif False: return "B"
elif points >= 70: return "C"
else: return "F"

```

points : 75

```

if False: return "A"
elif False: return "B"
elif True: return "C"
else: return "F"

```

points : 75

```
return "C"
```

points : 75

```
"C"
```

```

def my_abs(n):
    if n < 0:
        return n * -1
    else:
        return n

def letter_grade1(points):
    if points >= 90: return "A"
    elif points >= 80: return "B"
    elif points >= 70: return "C"
    else: return "F"

```

```

def letter_grade2(points):
    if points >= 90: return "A"
    elif points >= 80: return "B"
    elif points >= 70: return "C"
    elif points < 70: return "F"

```

```

def letter_grade3(points):
    if points < 70: return "F"
    elif points >= 70: return "C"
    elif points >= 80: return "B"
    elif points >= 90: return "A"

```

```

def letter_grade4(points):
    if points < 70: return "F"
    elif points < 80: return "C"
    elif points < 90: return "B"
    else: return "A"

```

```

# Write a function phase_of_water that takes a number
# representing degrees Celsius and returns "liquid",
# "solid", or "gas" depending on if water would be
# liquid, ice, or steam at that temperature.

```

### Example trace:

```
my_abs(7)
```

```

if n < 0: return n * -1
else: return n

```

n : 7

```

if False: return n * -1
else: return n

```

n : 7

```
return n
```

n : 7

```
7
```