Numeric data types

DATA TYPES FOR DATA SCIENCE IN PYTHON

Jason Myers
Instructor

datacamp

Built in numeric types

Integer

- Whole numbers
- Large values

```
int(12345678912345678 9)
```

```
12345678912345678 9
```

Float

- Fractional amounts (approximation)
- Scientific notation

```
float(12345678912345678 9)
```

```
1.2345678912345678e+17
```

Decimals

- Exact precision
- Currency operations

```
from decimal import Decimal
Decimal('123456789123456789')
```

```
Decimal('123456789123456789')
```

Printing floats

```
print(0.00001)
```

```
1e-05
```

```
print(f"{0.00001:f}")
```

```
0.000010
```

Printing floats

```
print(f"{0.0000001:f}")
```

```
0.000000
```

```
print(f"{0.0000001:.7f}")
```

```
0.0000001
```

Python division types

```
4/2
```

```
2.0
```

```
4//2
```

```
2
```

```
7//3
```

```
2
```

Let's practice!

DATA TYPES FOR DATA SCIENCE IN PYTHON

Booleans - the logical data type

DATA TYPES FOR DATA SCIENCE IN PYTHON

Jason Myers
Instructor

datacamp

Booleans as a data type

- `True`
- `False`

Notice the capitalization as this can trip you up when switching between Python and other languages.

```
out_of_cookies = True
if out_of_cookies:
    print("Run to the store NOW!")
```

```
Run to the store NOW!
```

Truthy and Falsey

- Truthy values are ones that will return true
- Falsey values will evaluate to false

```
apples=2
if apples:
    print("We have apples.")
```

```
"We have apples."
```

```
apples=0
if apple:
    print('We have apples.')
```

Truthy and Falsey

Truthy

- `1`
- `"Cookies"`
- `["Cake", "Pie"]`
- `{"key": "value"}`

Falsey

- `0`
- `""`
- `[]`
- `{}`
- `None`

Operators - a boolean evaluation context

```
cookie_qty == 3
```

- `==` equal to
- `!=` not equal to
- `<` less than
- `<=` less than or equal to
- `>` greater than
- `>=` greater than or equal to

Floats are approximately an issue

```
x = 0.1 + 1.1
x == 1.2
```

```
False
```

```
print(x)
```

```
1.2000000000000002
```

Be careful with equality comparisons of floats!

Let's practice!

DATA TYPES FOR DATA SCIENCE IN PYTHON

Sets (unordered data with optimized logic operations)

DATA TYPES FOR DATA SCIENCE IN PYTHON

Jason Myers
Instructor

datacamp

Set

- Unique
- Unordered
- Mutable
- Python's implementation of Set Theory from Mathematics

Creating sets

- Sets are created from a list

```
cookies_eaten_today = ['chocolate chip', 'peanut butter',
   ...: 'chocolate chip', 'oatmeal cream', 'chocolate chip']
types_of_cookies_eaten = set(cookies_eaten_today)
print(types_of_cookies_eaten)
```

```
set(['chocolate chip', 'oatmeal cream', 'peanut butter'])
```

Modifying sets

- `.add()` adds single elements

```
types_of_cookies_eaten.add('biscotti')

types_of_cookies_eaten.add('chocolate chip')

print(types_of_cookies_eaten)
```

```
set(['chocolate chip', 'oatmeal cream', 'peanut butter', 'biscotti'])
```

Updating sets

- `.update()` merges in another set or list

```
cookies_hugo_ate = ['chocolate chip', 'anzac']

types_of_cookies_eaten.update(cookies_hugo_ate)

print(types_of_cookies_eaten)
```

```
set(['chocolate chip', 'anzac', 'oatmeal cream',  'peanut
```

Removing data from sets

- `.discard()` safely removes an element from the set by value
- `.pop()` removes and returns an arbitrary element from the set (KeyError when empty)

```
types_of_cookies_eaten.discard('biscotti')
print(types_of_cookies_eaten)
```

```
set(['chocolate chip', 'anzac', 'oatmeal cream', 'peanut butter'])
```

```
types_of_cookies_eaten.pop()
types_of_cookies_eaten.pop()
```

```
'chocolate chip'
'anzac'
```

Set operations - similarities

- `.union()` set method returns a set of all the names (`or`)
- `.intersection()` method identifies overlapping data (`and`)

```
cookies_jason_ate = set(['chocolate chip', 'oatmeal cream',
'peanut butter'])
cookies_hugo_ate = set(['chocolate chip', 'anzac'])
cookies_jason_ate.union(cookies_hugo_ate)
```

```
set(['chocolate chip', 'anzac', 'oatmeal cream', 'peanut butter'])
```

```
cookies_jason_ate.intersection(cookies_hugo_ate)
```

```
set(['chocolate chip'])
```

Set operations - differences

- `.difference()` method identifies data present in the set on which the method was used that is not in the arguments (`-`)
- Target is important!

```
cookies_jason_ate.difference(cookies_hugo_ate)
```

```
set(['oatmeal cream', 'peanut butter'])
```

```
cookies_hugo_ate.difference(cookies_jason_ate)
```

```
set(['anzac'])
```

Let's practice!

DATA TYPES FOR DATA SCIENCE IN PYTHON