Lambda functions

PYTHON DATA SCIENCE TOOLBOX (PART 1)



Hugo Bowne-Anderson Instructor



Lambda functions

```
raise_to_power = lambda x, y: x ** y
raise_to_power(2, 3)
```

8

Anonymous functions

- Function map takes two arguments: map(func, seq)
- map() applies the function to ALL elements in the sequence

```
nums = [48, 6, 9, 21, 1]
square_all = map(lambda num: num ** 2, nums)
print(square_all)
```

```
<map object at 0x103e065c0>
```

```
print(list(square_all))
```

```
[2304, 36, 81, 441, 1]
```

Note

- map() and lambda functions

The function map() applies a function over an object, such as a list.

- filter() and lambda functions

The function filter() offers a way to filter out elements from a list that don't satisfy certain criteria.

- reduce() and lambda functions

The reduce() function is useful for performing some computation on a list and, unlike map() and filter(), returns a single value as a result. To use reduce(), you must import it from the functools module.

Import reduce from functools

Let's practice!

PYTHON DATA SCIENCE TOOLBOX (PART 1)



Introduction to error handling

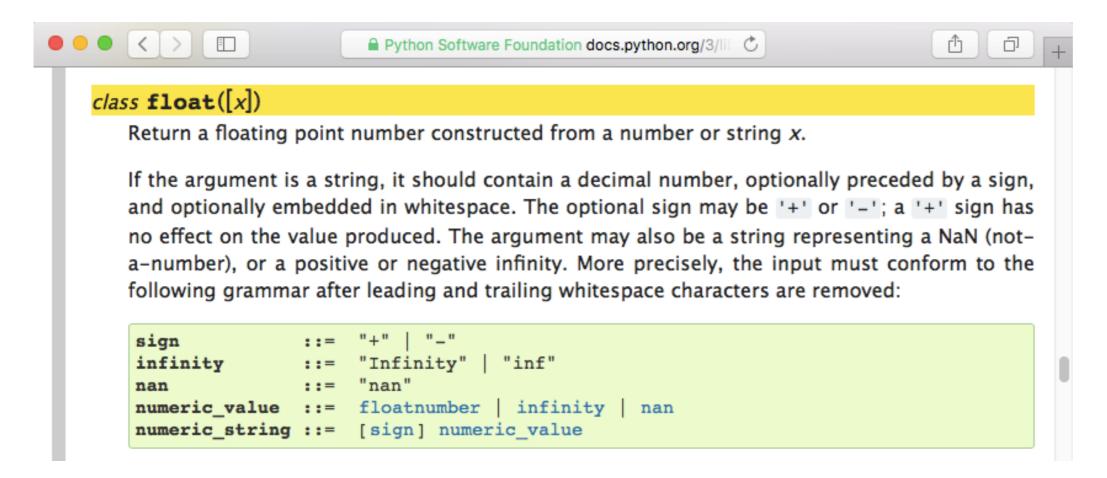
PYTHON DATA SCIENCE TOOLBOX (PART 1)



Hugo Bowne-Anderson
Instructor



The float() function



Passing an incorrect argument

```
float(2)
2.0
float('2.3')
2.3
float('hello')
                                 Traceback (most recent call last)
ValueError
<ipython-input-3-d0ce8bccc8b2> in <module>()
<hr />-> 1 float('hi')
ValueError: could not convert string to float: 'hello'
```



Passing valid arguments

```
def sqrt(x):
    """Returns the square root of a number."""
    return x ** (0.5)
sqrt(4)
```

2.0

sqrt(10)

3.1622776601683795



Passing invalid arguments

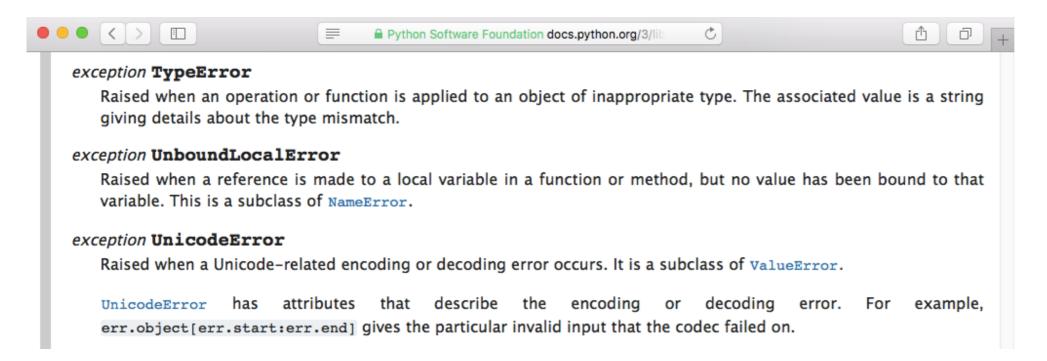


- Exceptions caught during execution
- Catch exceptions with try-except clause
 - Runs the code following try
 - If there's an exception, run the code following except

```
def sqrt(x):
    """Returns the square root of a number."""
   try:
       return x ** 0.5
   except:
       print('x must be an int or float')
sqrt(4)
sqrt(10.0)
3.1622776601683795
sqrt('hi')
x must be an int or float
```



```
def sqrt(x):
    """Returns the square root of a number."""
    try:
        return x ** 0.5
    except TypeError:
        print('x must be an int or float')
```



```
sqrt(-9)
```

```
(1.8369701987210297e-16+3j)
```

```
def sqrt(x):
    """Returns the square root of a number."""
    if x < 0:
        raise ValueError('x must be non-negative')
    try:
        return x ** 0.5
    except TypeError:
        print('x must be an int or float')</pre>
```

```
sqrt(-2)
```



Let's practice!

PYTHON DATA SCIENCE TOOLBOX (PART 1)



Bringing it all together

PYTHON DATA SCIENCE TOOLBOX (PART 1)



Hugo Bowne-Anderson
Instructor



```
def sqrt(x):
    try:
        return x ** 0.5
    except:
        print('x must be an int or float')
sqrt(4)
2.0
sqrt('hi')
x must be an int or float
```



```
def sqrt(x):
    if x < 0:
        raise ValueError('x must be non-negative')
    try:
        return x ** 0.5
    except TypeError:
        print('x must be an int or float')</pre>
```

Let's practice!

PYTHON DATA SCIENCE TOOLBOX (PART 1)



Congratulations!

PYTHON DATA SCIENCE TOOLBOX (PART 1)



Hugo Bowne-Anderson
Instructor



What you've learned:

- Write functions that accept single and multiple arguments
- Write functions that return one or many values
- Use default, flexible, and keyword arguments
- Global and local scope in functions
- Write lambda functions
- Handle errors

There's more to learn!

- Create lists with list comprehensions
- Iterators you've seen them before!
- Case studies to apply these techniques to Data Science

Let's practice!

PYTHON DATA SCIENCE TOOLBOX (PART 1)

