Functions

INTRODUCTION TO PYTHON

Hugo Bowne-Anderson

Da a S ce a Da a Ca

Functions

- N h g e
- `type()`
- P e œ f e abe c dœ
- S e a c a a
- Ca f c e ad f g c dœ e

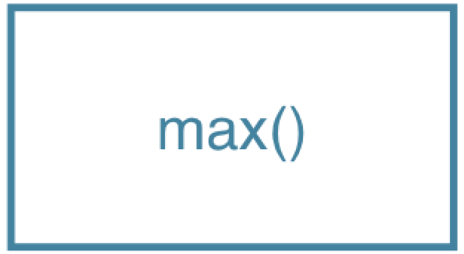
Example

```
fam = [1.73, 1.68, 1.71, 1.89]
fam
```

```
[1.73, 1.68, 1.71, 1.89]
```

```
max(fam)
```

```
1.89
```



max()

Example

```
fam = [1.73, 1.68, 1.71, 1.89]
fam
```

```
[1.73, 1.68, 1.71, 1.89]
```

```
max(fam)
```

```
1.89
```

[1.73, 1.68, 1.71, 1.89] → max()

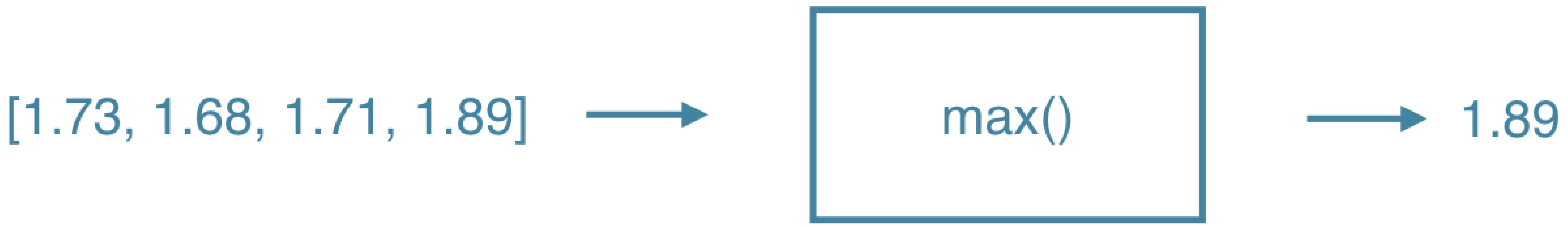
Example

```python
fam = [1.73, 1.68, 1.71, 1.89]
fam
```

```
[1.73, 1.68, 1.71, 1.89]
```

```python
max(fam)
```

```
1.89
```



[1.73, 1.68, 1.71, 1.89] → max() → 1.89

Example

```
fam = [1.73, 1.68, 1.71, 1.89]
fam
```

```
[1.73, 1.68, 1.71, 1.89]
```

```
max(fam)
```

```
1.89
```

```
tallest = max(fam)
tallest
```

```
1.89
```

round()

```python
round(1.68, 1)
```

```
1.7
```

```python
round(1.68)
```

```
2
```

```python
help(round) # Open up documentation
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round()

```
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round()

round()

```
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round()

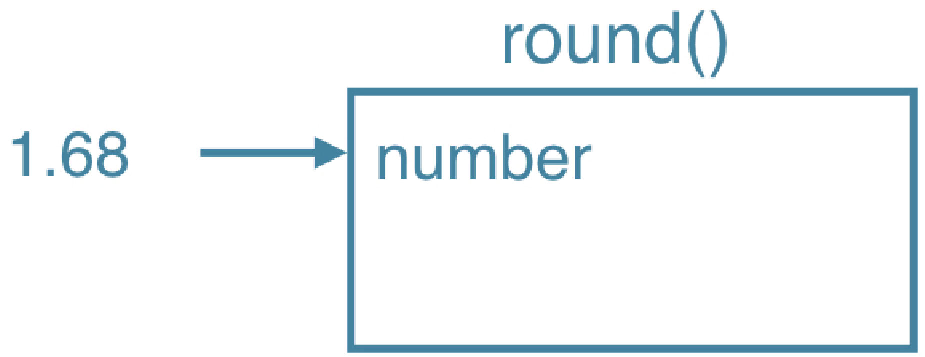
```
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round(1.68, 1)



round()

1.68

number

round()

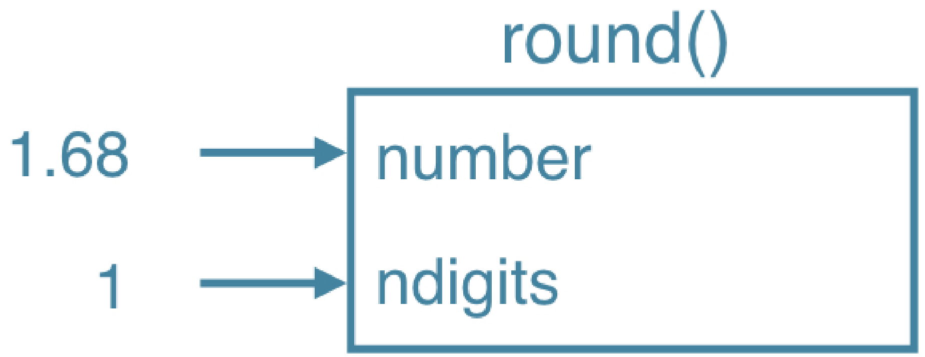
```
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round(1.68, 1)



round()

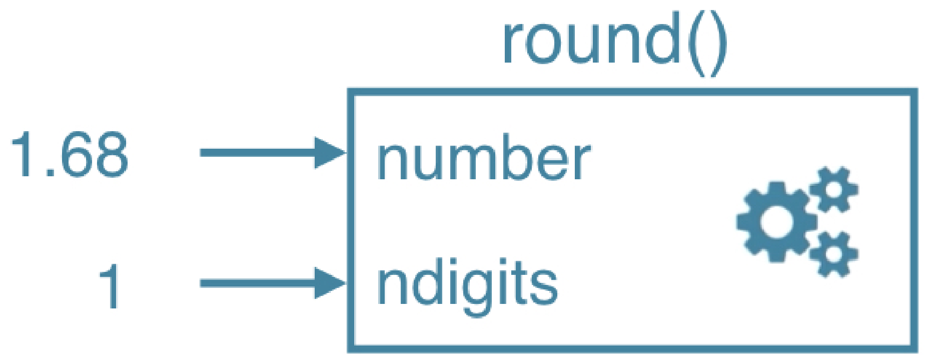
```
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round(1.68, 1)



round()

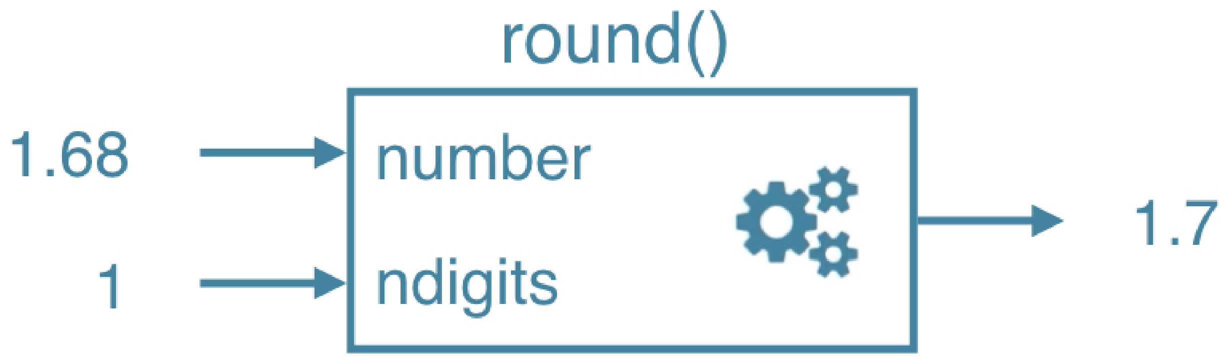
```
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round(1.68, 1)



round()

```python
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round()

```python
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round()

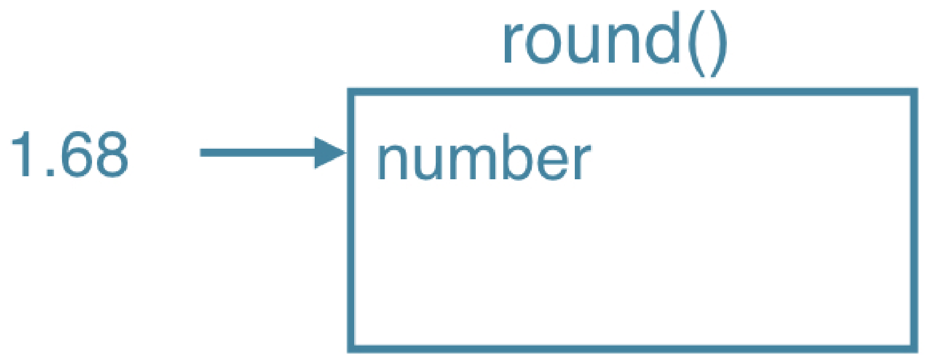
```python
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round(1.68)



round()

1.68

number

round()

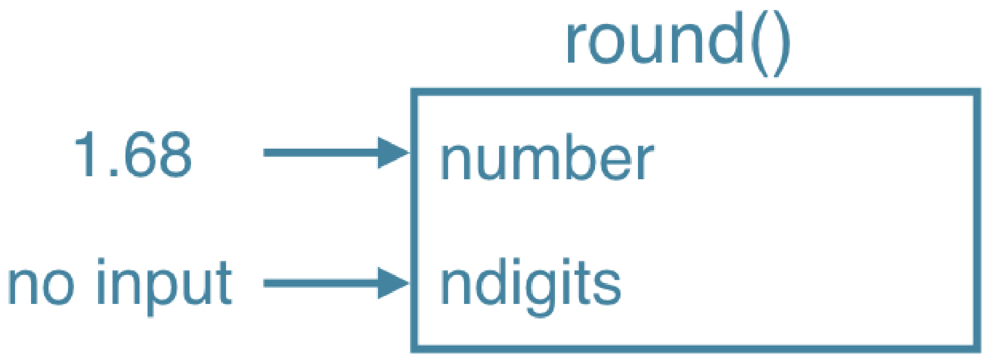
```
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round(1.68)



round()

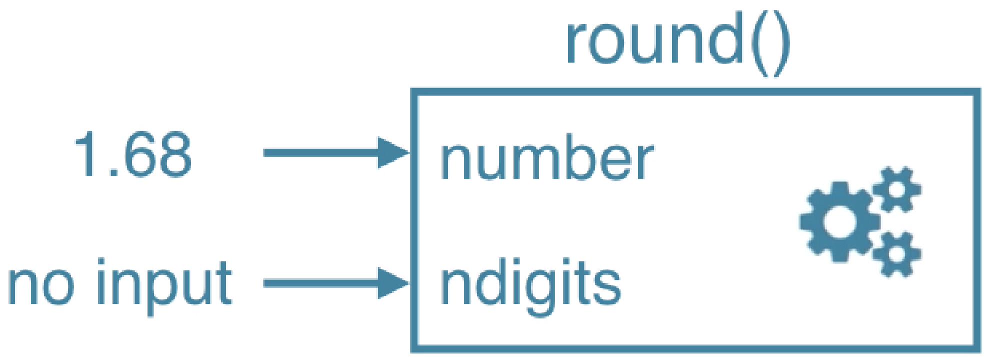
```
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round(1.68)



round()

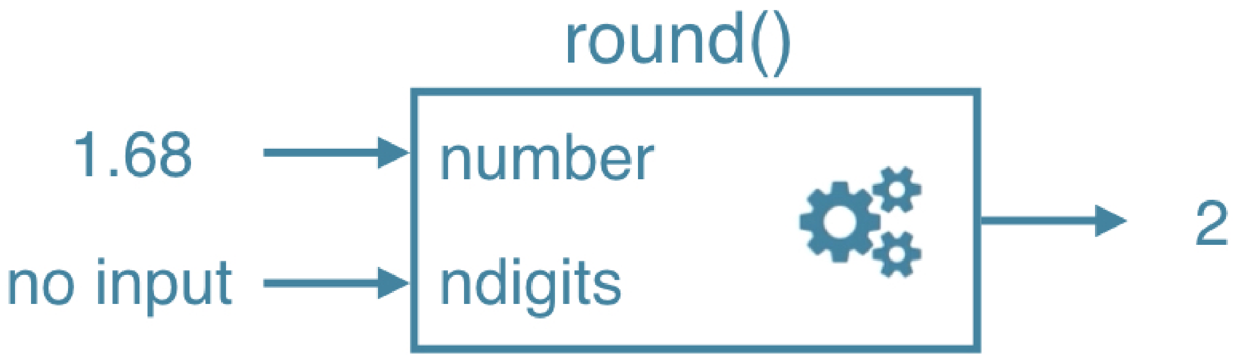
```python
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

round(1.68)



round()

```
help(round)
```

```
Help on built-in function round in module builtins:

round(number, ndigits=None)
    Round a number to a given precision in decimal digits.

    The return value is an integer if ndigits is omitted or None.
    Otherwise the return value has the same type as the number. ndigits may be negative.
```

- `round(number)`
- `round(number, ndigits)`

Find functions

- H
- S a dada d a bab f c e
- The e e f e d

Let's practice!

INTRODUCTION TO PYTHON

Methods

INTRODUCTION TO PYTHON

Hugo Bowne-Anderson

Data Scientist at DataCamp

Built-in Functions

- Ma f a
- Le gh f g e
- Ge de
- Re e g a

Back 2 Basics

```
sister = "liz"
```

Object

```
height = 1.73
```

Object

```
fam = ["liz", 1.73, "emma", 1.68,
       "mom", 1.71, "dad", 1.89]
```

Object

Back 2 Basics

```python
sister = "liz"
```

```python
height = 1.73
```

```python
fam = ["liz", 1.73, "emma", 1.68,
       "mom", 1.71, "dad", 1.89]
```

	type
Object	str
Object	float
Object	list

- Meh d F c h a
  be g bec

Back 2 Basics

```
sister = "liz"
```

```
height = 1.73
```

```
fam = ["liz", 1.73, "emma", 1.68,
       "mom", 1.71, "dad", 1.89]
```

- Meh d F c h a be g bec

	type	examples of methods

Object	str	capitalize()
		replace()

Object	float	bit_length()
		conjugate()

Object	list	index()
		count()

list methods

```
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

```
fam.index("mom") # "Call method index() on fam"
```

```
4
```

```
fam.count(1.73)
```

```
1
```

str methods

Methods

- E e h g bec
- Obec h a e eh d a caed dœ e d g e

```
sister.replace("z", "sa")
```

```
'lisa'
```

```
fam.replace("mom", "mommy")
```

```
AttributeError: 'list' object has no attribute 'replace'
```

Methods

```python
sister.index("z")
```

```
2
```

```python
fam.index("mom")
```

```
4
```

Methods (2)

```
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89]
```

```
fam.append("me")
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89, 'me']
```

```
fam.append(1.79)
fam
```

```
['liz', 1.73, 'emma', 1.68, 'mom', 1.71, 'dad', 1.89, 'me', 1.79]
```

Summary

F c

```
type(fam)
```

```
list
```

Me h d ca f c be c

```
fam.index("dad")
```

```
6
```

Let's practice!

INTRODUCTION TO PYTHON

Packages

INTRODUCTION TO PYTHON

Hugo Bowne-Anderson

Da aS ce a Da aCa

Motivation

- F c a d e h d ae e f
- A c dœ P h d b
  - H gœ c dœ bae e
  - L f c dœ e
  - Ma e a œ be

Packages

- Dec P h Sc
- Each c d e
- S ecf f c eh d e
- Th a d f ac age a a abe
    - N P
    - Ma b
    - c ea

```
pkg/
  mod1.py
  mod2.py
  ...
```

Install package

- http://pip.readthedocs.org/en/stable/installing/
- D ad`get-pip.py`
- Te a
  - `python3 get-pip.py`
  - `pip3 install numpy`

Import package

```
import numpy
array([1, 2, 3])
```

```
NameError: name 'array' is not defined
```

```
numpy.array([1, 2, 3])
```

```
array([1, 2, 3])
```

```
import numpy as np
np.array([1, 2, 3])
```

```
array([1, 2, 3])
```

```
from numpy import array
array([1, 2, 3])
```

```
array([1, 2, 3])
```

from numpy import array

- `my_script.py`

```
from numpy import array

fam = ["liz", 1.73, "emma", 1.68,
    "mom", 1.71, "dad", 1.89]


...
fam_ext = fam + ["me", 1.79]


...
print(str(len(fam_ext)) + " elements in fam_ext")


...
np_fam = array(fam_ext)
```

- U g N P b e ce a

import numpy

```
import numpy as np

fam = ["liz", 1.73, "emma", 1.68,
    "mom", 1.71, "dad", 1.89]


...
fam_ext = fam + ["me", 1.79]


...
print(str(len(fam_ext)) + " elements in fam_ext")


...
np_fam = np.array(fam_ext) # Clearly using NumPy
```

Let's practice!

INTRODUCTION TO PYTHON