



DSC 10, Spring 2018

Lecture 3

Data Types

sites.google.com/eng.ucsd.edu/dsc-10-spring-2018

Announcements

- HW 1 due Sunday 11:59pm
- Guest Speaker in Tuesday's Discussion: Dr. Aaron Fraenkel, Sr. Machine Learning Scientist at Amazon

Bias, Fairness, and Interpretability in Machine Learning

“Machine Learning occasionally makes news for producing results perceived as biased, unfair, or just fantastically wrong. We'll examine why these results aren't mysterious or unexpected, as long as you ask the right questions and know the data driving your results.”

Call Expressions

(Demo)

Anatomy of a Call Expression

function (operation)

`abs(-12)`

argument (input)

Anatomy of a Call Expression

What
function
to call

How to compute
the first
argument

How to compute
the second
argument

f (**x + y** , **g (z)**)

"Call f on the result of adding x + y and the return value of calling g on z."

Discussion Question

Assume you have run the following statements:

$$x = 3$$
$$y = -2$$

Which of these examples results in an error?

- A. `abs(x, y)`
- B. `math.pow(x, abs(y))`
- C. `round(x, max(abs(y**2)))`
- D. `math.pow(x, math.pow(y, x))`
- E. More than one of the above

Strings

Text, Strings, and Types

A string value is a snippet of text of any length

- `'a'`
- `'word'`
- `"there can be 2 sentences. Here's the second!"`

Text, Strings, and Types

A string value is a snippet of text of any length

- `'a'`
- `'word'`
- `"there can be 2 sentences. Here's the second!"`

Strings that contain numbers can be converted to numbers

- `int('12')`
- `float('1.2')`

Text, Strings, and Types

A string value is a snippet of text of any length

- `'a'`
- `'word'`
- `"there can be 2 sentences. Here's the second!"`

Strings that contain numbers can be converted to numbers

- `int('12')`
- `float('1.2')`

Any value can be converted to a string

- `str(5)`

(Demo)

Discussion Question

Assume you have run the following statements.

```
x = 3
```

```
y = '4'
```

```
z = '5.6'
```

Choose the expression that will be evaluated without an error.

A. `x + y`

B. `x + int(y + z)`

C. `str(x) + int(y)`

D. `str(x) + z`

E. All of them have errors

Arrays

Arrays

An array contains a sequence of values

Arrays

An array contains a sequence of values

- All elements of an array should have the same type

Arrays

An array contains a sequence of values

- All elements of an array should have the same type
- Arithmetic is applied to each element individually

Arrays

An array contains a sequence of values

- All elements of an array should have the same type
- Arithmetic is applied to each element individually
- When two arrays are added, they must have the same size; corresponding elements are added in the result

Arrays

An array contains a sequence of values

- All elements of an array should have the same type
- Arithmetic is applied to each element individually
- When two arrays are added, they must have the same size; corresponding elements are added in the result

<http://berkeleyearth.lbl.gov/regions/global-land>

(Demo)

Ranges

Ranges

A range is an array of consecutive numbers

Ranges

A range is an array of consecutive numbers

- **`np.arange(end)`** :

An array of increasing integers from 0 up to **`end`**

Ranges

A range is an array of consecutive numbers

- **`np.arange(end)`** :
An array of increasing integers from 0 up to **`end`**
- **`np.arange(start, end)`** :
An array of increasing integers from **`start`** up to **`end`**

Ranges

A range is an array of consecutive numbers

- `np.arange(end)`:
An array of increasing integers from 0 up to `end`
- `np.arange(start, end)`:
An array of increasing integers from `start` up to `end`
- `np.arange(start, end, step)`:
A range with `step` between consecutive values

Ranges

A range is an array of consecutive numbers

- `np.arange(end):`

An array of increasing integers from 0 up to **end**

- `np.arange(start, end):`

An array of increasing integers from **start** up to **end**

- `np.arange(start, end, step):`

A range with **step** between consecutive values

A range always includes **start** but excludes **end**

[,)

(Demo)

Discussion Question

Assume you have run the following statements.

```
x = make_array(2, 3, 4)
y = np.arange(2, 3, 4)
z = np.arange(3)
```

Which line(s) will cause an error?

A. `x + y`

B. `x + z`

C. `x.item(0) + y.item(0)`

D. `x.item(1) + y.item(1)`

Leibniz Formula for Pi

$$\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots\right)$$

```
a = np.arange(1, 12, 4)
```

```
a
```

```
a + 2
```

```
4 * sum(1/a - 1/(a+2))
```

Leibniz Formula for Pi

$$\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots\right)$$

```
a = np.arange(1, 12, 4)
```

```
a
```

```
a + 2
```

```
4 * sum(1/a - 1/(a+2))
```

Which of the following is false?

- A. `sum` is a function being applied to an array
- B. `a` is an array
- C. `a` is a range
- D. The last line is equivalent to

```
4 * (1/a + -1/(a+2))
```

(Demo)