Lecture 4
Data Types

Contributions by Vinitra Swamy (vinitra@berkeley.edu) and Fahad Kamran (fhdkmrn@berkeley.edu)
Slides created by John DeNero (denero@berkeley.edu) and Ani Adhikari (adhikari@berkeley.edu)
Announcements
Python Programming Language

- Table structure within the `datascience` package
- CA minimum wage, ice-cream cones, NBA player statistics

<table>
<thead>
<tr>
<th>Name</th>
<th>Code</th>
<th>Area (m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>California</td>
<td>CA</td>
<td>163696</td>
</tr>
<tr>
<td>Nevada</td>
<td>NV</td>
<td>110567</td>
</tr>
</tbody>
</table>
What are the column labels of each table?

```python
x = cones.select('Flavor', 'Color')
x
y = x.drop('Color')
y
x = cones.select('Color', 'Price')
x
y
```
Arithmetic
# Arithmetic Operators

<table>
<thead>
<tr>
<th>Operation</th>
<th>Operator</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>+</td>
<td>2 + 3</td>
<td>5</td>
</tr>
<tr>
<td>Subtraction</td>
<td>-</td>
<td>2 - 3</td>
<td>-1</td>
</tr>
<tr>
<td>Multiplication</td>
<td>*</td>
<td>2 * 3</td>
<td>6</td>
</tr>
<tr>
<td>Division</td>
<td>/</td>
<td>7 / 3</td>
<td>2.66667</td>
</tr>
<tr>
<td>Remainder</td>
<td>%</td>
<td>7 % 3</td>
<td>1</td>
</tr>
<tr>
<td>Exponentiation</td>
<td>**</td>
<td>2 ** 0.5</td>
<td>1.41421</td>
</tr>
</tbody>
</table>
Python has two real number types

- **int**: an integer of any size
- **float**: a number with an optional fractional part

An **int** never has a decimal point; a **float** always does.

A **float** might be printed using scientific notation.

Three limitations of float values:

- They have limited size (but the limit is huge)
- They have limited precision of 15-16 decimal places
- After arithmetic, the final few decimal places can be wrong
Rank the results of the following expressions in order from least to greatest

A. \( 3 \times 10^{10} \)
B. \( 10 \times 3^{10} \)
C. \( (10 \times 3)^{10} \)
D. \( 10 / 3 / 10 \)
E. \( 10 / (3 / 10) \)

A. 30000000000
B. 590490
C. 590490000000000
D. 0.33333333333333337
E. 33.333333333333336
Strings
Text and Strings

A string value is a snippet of text of any length
- 'a'
- 'word'
- "there can be 2 sentences. Here's the second!"

Strings consisting of numbers can be converted to numbers
- int('12')
- float('1.2')

Any value can be converted to a string
- str(5)
Discussion Question

Assume you have run the following statements

\[
x = 3 \\
y = '4' \\
z = '5.6'
\]

What's the source of the error in each example?

A. \( x + y \)
B. \( x + \text{int}(y + z) \)
C. \( \text{str}(x) + \text{int}(y) \)
D. \( \text{str}(x, y) + z \)
Types
Every value has a type

We’ve seen 5 types so far:

- int: 2
- float: 2.2
- str: 'Red fish, blue fish'

The `type` function can tell you the type of a value

- `type(2)`
- `type(2 + 2)`

An expression’s “type” is based on its value, not how it looks

- `x = 2`
- `type(x)`

(Demo)
Conversions

Strings that contain numbers can be converted to numbers

- `int('12')`
- `float('1.2')`
- `float('one point two')` # Not a good idea!

Any value can be converted to a string

- `str(5)`

Numbers can be converted to other numeric types

- `float(1)`
- `int(1.2)` # DANGER: loses information!
Arrays
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
- A column of a table is an array

(Demo)