Lecture 4
Data Types
Announcements

- Website: [cornell-dsfa.org](http://cornell-dsfa.org).
- If you are just joining...
- Reminder: HW 1 out, due Friday by 6PM, bonus point for turn-in on Thursday.
  - Need help? See office hours in Zoom, and Ed Discussions via Canvas.
Announcements

- If you want to follow along for the lecture notebook, go to Canvas->Assignments->Lecture Demos.
  - If lec04.ipynb not there, try clicking “Actions” (upper right corner), “Reset assignment”. Note: this will erase (I think) whatever you had in previous lecture demo notebooks.
  - Note: You need to run the first cell in the demo!
Tables
Table Structure

- We organize our data in tables
- A Table is a sequence of labeled columns
- Data within a column should be of the same "type"

<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>
Table Operations

- \texttt{t.select(label)} - constructs a new table with just the specified columns
- \texttt{t.sort(label)} - constructs a new table, with rows sorted by the specified column
Visualization

- `t.barh(label)` - horizontal bar chart with specified column as the y-axis categories

(Demo)
Table Operations

- `t.where(label, condition)` - constructs a new table with just the rows that match the condition

(Demo)
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>
PEMDAS

( ) x² x or ÷ + or -

LEFT TO RIGHT LEFT TO RIGHT

Parentheses Exponents
Multiplication Division
Addition Subtraction

Penguins Eat Many Donuts After School!
Python has two numeric 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
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 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

\[
\begin{align*}
x &= 3 \\
y &= '4' \\
z &= '5.6'
\end{align*}
\]

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. \( \text{str}(x, y) + z \)
Arrays and Ranges
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)
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

The range always includes `start` but excludes `end`