Part II: Python
Growth of major programming languages
Based on Stack Overflow question views in World Bank high-income countries

Stackoverflow blog, September 2017
Year over year growth in traffic to programming languages/platforms

Comparing question views in January-August of 2016 and 2017, in World Bank high-income countries. TypeScript had a growth rate of 142% and an average size of 0.36%, and was omitted.
Python has many applications

• Web development
• Application development
• Computer graphics
• Scientific computing
  – Bioinformatics
  – Machine learning
  – Simulations

https://www.python.org/about/quotes/
Three alternatives to get Python

• Jupyterhub on educcomp (in browser)
• Google Colaboratory (in browser)  
  https://colab.research.google.com/
• Anaconda (local install, ~1.5GB of space required)
Please choose one of the following applications:

- RStudio
- Jupyterhub
Select items to perform actions on them.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **R**: a month ago
- **worksheets**: a month ago
In [1]: print("Hello World!")

Hello World!

In [ ]:
Counting like a computer scientist

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ...
Indexing in Python

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>y</td>
<td>t</td>
<td>h</td>
<td>o</td>
<td>n</td>
</tr>
</tbody>
</table>
Indexing in Python

In [1]: x="Python"

In [2]: x[0]
Out[2]: 'P'
Indexing in Python

In [1]: x="Python"

In [2]: x[1:4] ← We index from the first element to one past the last element

Out[2]: 'yth'
Indexing in Python

```
In [1]: x="Python"
In [2]: x[3:]  # Missing number means “to the end"
Out[2]: 'hon'
```
We can also index in reverse

<table>
<thead>
<tr>
<th>P</th>
<th>y</th>
<th>t</th>
<th>h</th>
<th>o</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
</tbody>
</table>
We can also index in reverse

<table>
<thead>
<tr>
<th>P</th>
<th>y</th>
<th>t</th>
<th>h</th>
<th>o</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>–6</td>
<td>–5</td>
<td>–4</td>
<td>–3</td>
<td>–2</td>
<td>–1</td>
</tr>
</tbody>
</table>

In [1]: `x="Python"`

In [2]: `x[-6]`

Out[2]: 'P'
We can also index in reverse

<table>
<thead>
<tr>
<th>P</th>
<th>y</th>
<th>t</th>
<th>h</th>
<th>o</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>-6</td>
<td>-5</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
</tbody>
</table>

In [1]: x="Python"

In [2]: x[−5:−2] ← Again, we index one past the last element

Out[2]: 'yth'
We can also index in reverse

In [1]: x="Python"

In [2]: x[−3:]  This captures the last 3 characters
Out[2]: 'hon'