

University of Toronto **Engineering**



UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Module 7 – Python

Programming for the web



UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Module 7 – Python

Python is a high-level, structured language

- It is 3rd-generation – it contains structured code
- It is modular – we can import modules for almost anything!
- It lets us dynamically create a web page's content



Module 7 – Python

Let's try it:

- In the EC2 or idle, type “python” at the command prompt
- You'll see >>>
- Enter something!
 - *It'll probably throw an error*
- Enter ‘print “Hello World!”’
- Try “1+1”
- Let's set a variable!
 - $x=4$
 - *print x (or just x)*
 - $x*2$



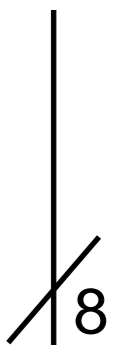
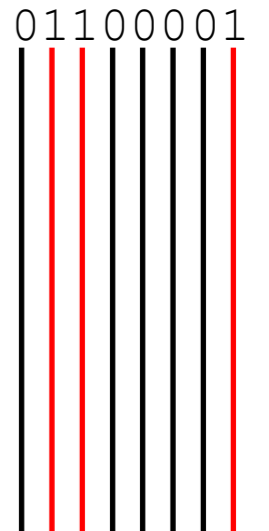
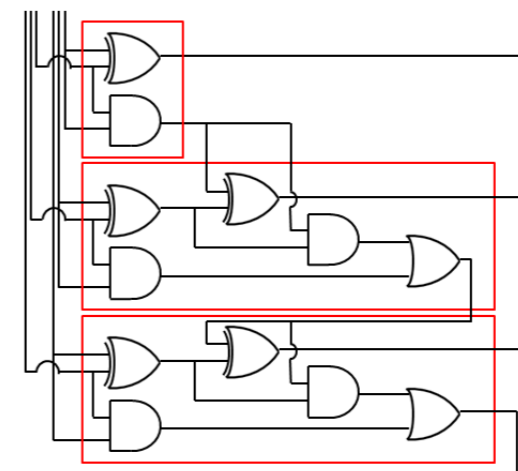
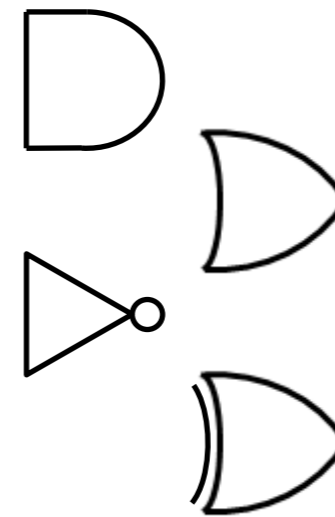
Module 7 – Python

The bitwise operators work on the binary forms of numbers

- &
- |
- ~
- ^
- >>
- <<

The arithmetic operators work on the numerical values

- +, -, *, /, **
- % - modulus



0b01100001
OR
0d97



Module 7 – Python

Boolean values can only be true or false

- Try making a boolean variable!
 - `y=True` *#The capital T is important!*
 - `z=False`
- A true is a 1, a false is a 0!
- We can apply our logical gates to python boolean values too!
 - `y and z`
 - `y or 0`
 - `not y`
- We can generate booleans with comparison operators:
 - `==, !=, <, >, <=, >=`



Module 7 – Python

We can use boolean values for if statements:

- `if z==1:`
- `print("text!")`
- `<enter>`

Experiment with this...

- Note that the indentation is important
 - *It's not like HTML*



Module 7 – Python

Strings hold text!

- Use ‘single’ or “double” quotation marks
- “““triple””” quotation marks allow a string to span multiple lines
- To include a literal quotation mark, put a backslash in front of it: “\””=
- Try multiplying and adding strings: ‘D’+’E’*2+’P’
- Square brackets get a certain character: ‘abcdef’[3]=‘d’ (numbering starts at 0)
- Use a:b to get a substring: ‘abcdef’[1:3]=‘bc’ (this is not inclusive of the latter number)



Module 7 – Python

Strings are like a list of characters...

- We can have lists of other stuff too!
 - *`list=[1,2,3,4,5]` `#Use square brackets to make one`*
- You can add (+) lists, and multiply them by integers; you can index them (`[1,2,3,4,5][4]`) and get slices (`[1,2,3,4,5][1:4]`) just like strings!
 - *You can even assign to slices: `list[1]=6`*
 - *`list[2:4]=[0,0,0]`*
 - *`print(list)`*
 - *`list.append(5)` adds 5 to the end!*
 - *`[]` is the empty list*
- The `range()` function creates a list of consecutive numbers



Module 7 – Python

Loops repeat code

- for i in list:
 - `print(i)`

- `f=True`
- `i=2`
- while f:
 - `print(i)`
 - `i+=1`
 - `if i==9:`
 - `f=False`

- You can put loops inside loops, loop over ranges, over strings, etc etc... a prize for the coolest thing a loop can do!



Module 7 – Python

- Let's use python on the web!
 - *wget*
<https://bootstrap.pypa.io/get-pip.py>
 - *python get-pip.py*
 - *pip install web.py*
- Create a python script
 - *Put this basic code in it:*
- Run it!
 - *python w.py 80*

```
import web

urls = (
    '/', 'index`
    '/text.html', 'text'
)

class index:
    def GET(self):
        return "Hello, world!"

class text:
    def GET(self):
        return "Hello, world other text!"

if __name__ == "__main__":
    app = web.application(urls, globals())
    app.run()
```



Module 7 – Python

- Python can handle different URLs:

```
import web

urls = (
    '/', 'index',
    '/text', 'text',
    '/print/(.*)', 'print_txt'
)

class index:
    def GET(self):
        return "Hello, world!"

class print_txt:
    def GET(self,txt):
        return "Here's your text:
{0}".format(txt)

if __name__ == "__main__":
    app = web.application(urls, globals())
    app.run()
```



Module 7 – Python

- Python can handle form GET and POST arguments

```
import web

urls = (
    '/', 'index',
    '/print.html', 'print_txt'
)

class index:
    def GET(self):
        return """
<html>
<body>
<form action='/print.html'>
    <input type='text' name='q' />
    <input type='submit' />
</form>
</body>
</html>
"""

class print_txt:
    def GET(self,q):
        return "Your query was {0}".format(web.input().q)

if __name__ == "__main__":
    app = web.application(urls, globals())
    app.run()
```

- Go make something cool with this...



Thanks!

Questions?

Email: b.graydon@ieee.org



UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING