Python for Beginners

bo@oyoclass.com

Why Python?

- Python is a programming language
- Easy to learn, a lot of top universities (MIT, UC Berkeley) use Python to teach “Programming 101”
- Not a toy, quite powerful. From basic tools to web development, data analytics, etc, Python can handle all of them. Giant company love it too, like Google and NASA
- Write once, run everywhere. Windows, Mac OS, Unix/ Linux

Say Hello

- **print** “hello python”

Print Number

- **print** 5
- **print** “5”
- **print** 5 + 3
- **print** “5 + 3”

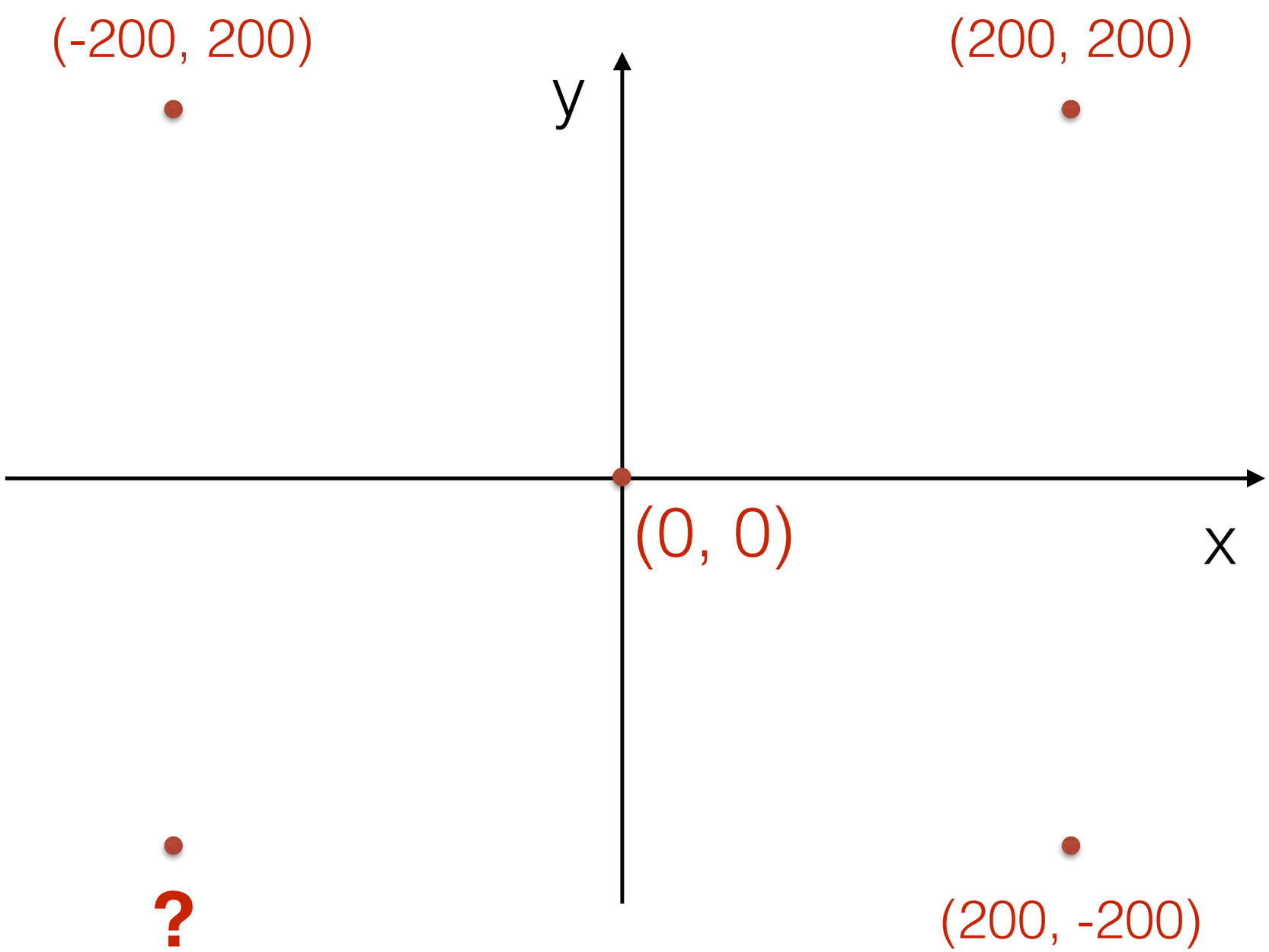
Calculator

- Addition: **+** , e.g. 1 + 1
- Substraction: **-** , e.g. 2 - 1
- Multiplication: ***** , e.g. 2 * 3
- Division: **/** , e.g. 6 / 3
- **print** 1 + 1
- **print** 1235 * 5678
- **print** 1 + 2 * 3
- **print** (1 + 2) * 3

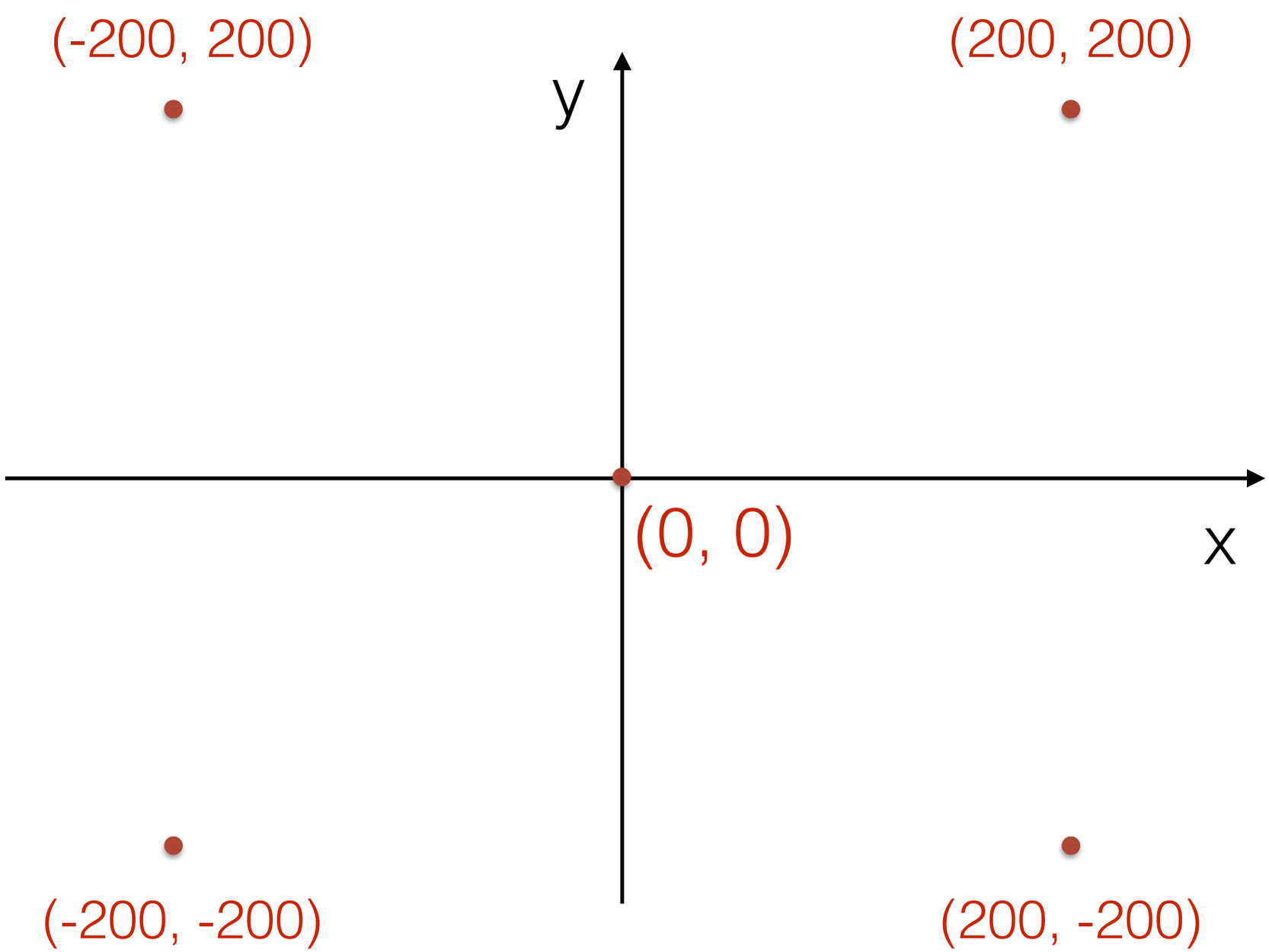
Python Turtle

```
# Import Turtle from library
from turtle import Turtle

# Generate your own turtle, give it a name
nick = Turtle()

# Control your turtle to move
nick.forward(50)
```

Turtle Position



Turtle Position



Turtle motion

move forward, e.g. nick.forward(50)
yourTurtleName.**forward**(*distance*)

move backward, e.g. nick.backward(50)
yourTurtleName.**backward**(distance)

turn right with certain degree, e.g. nick.right(90)
yourTurtleName.**right**(*angle*)

turn left with certain degree, e.g. nick.left(90)
yourTurtleName.**left**(*angle*)

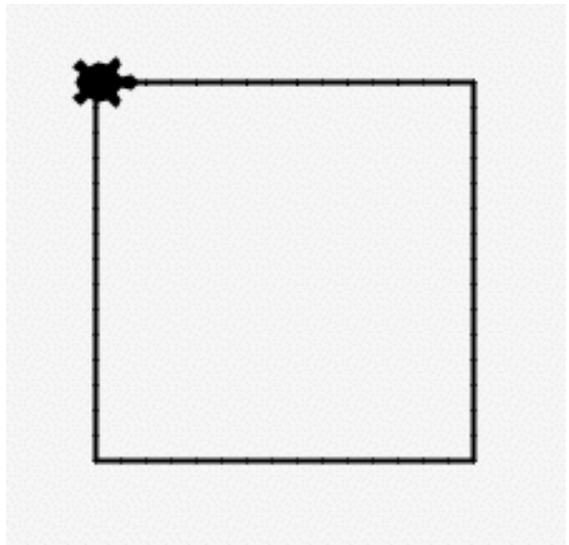
set turtle's coordinate, e.g. nick.setposition(10, 10)
yourTurtleName.**setposition**(*x, y*)

draw a circle with certain radius, e.g. nick.circle(20)
yourTurtleName.**circle**(*radius*)

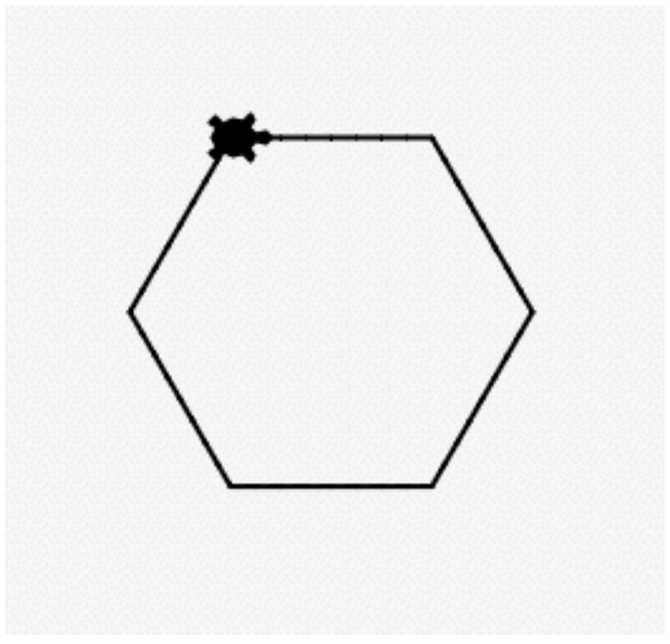
Turtle Appearance

```
# change shape to a real turtle
yourTurtleName.shape("turtle")

# stamp a copy of the turtle shape
yourTurtleName.stamp()

# change draw color, e.g. nick.color("red")
yourTurtleName.color(color)

# change line width, e.g. nick.width(5)
yourTurtleName.width(width)
```

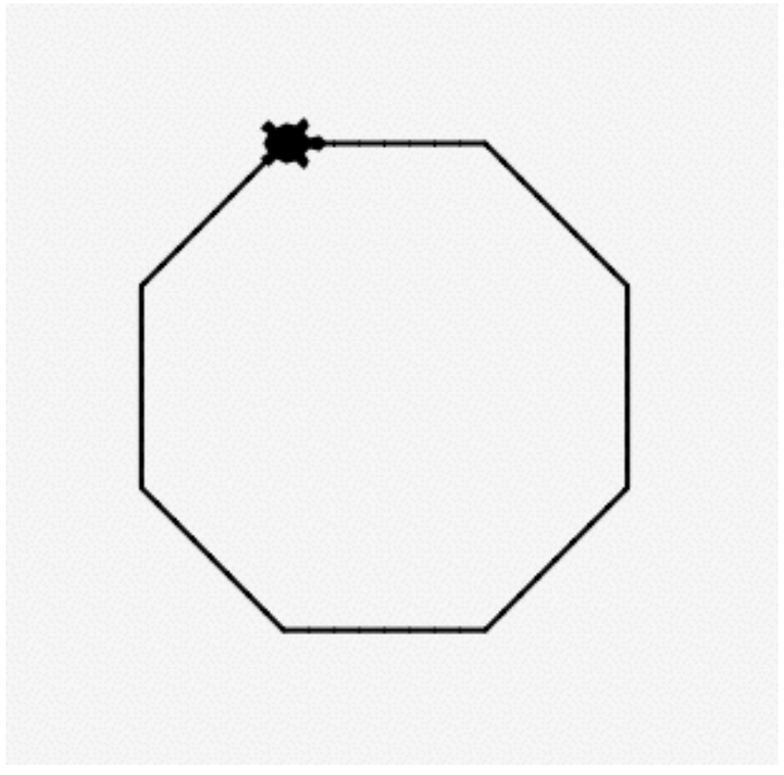
Draw a Square



Challenge: draw a hexagon



How about Octagon



```
nick.forward(80)
nick.right(60)
nick.forward(80)
nick.right(60)
nick.forward(80)
nick.right(60)
nick.forward(80)
nick.right(60)
nick.forward(80)
nick.right(60)
```

```
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
```

```
nick.forward(80)
nick.right(60)
nick.forward(80)
nick.right(60)
nick.forward(80)
nick.right(60)
nick.forward(80)
nick.right(60)
nick.forward(80)
nick.right(60)
nick.forward(80)
nick.right(60)
```

```
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
nick.forward(80)
nick.right(45)
```

Repeat / Loop:
let computer do the dirty work

```
for counter in [0, 1, 2, 3, 4]:
    print counter
    print “hello”
```

```
for counter in [0, 1, 2, 3, 4]:
    print counter
    print “hello”
```

Code Block

Indent

Repeat / Loop:
ask computer to do the dirty work

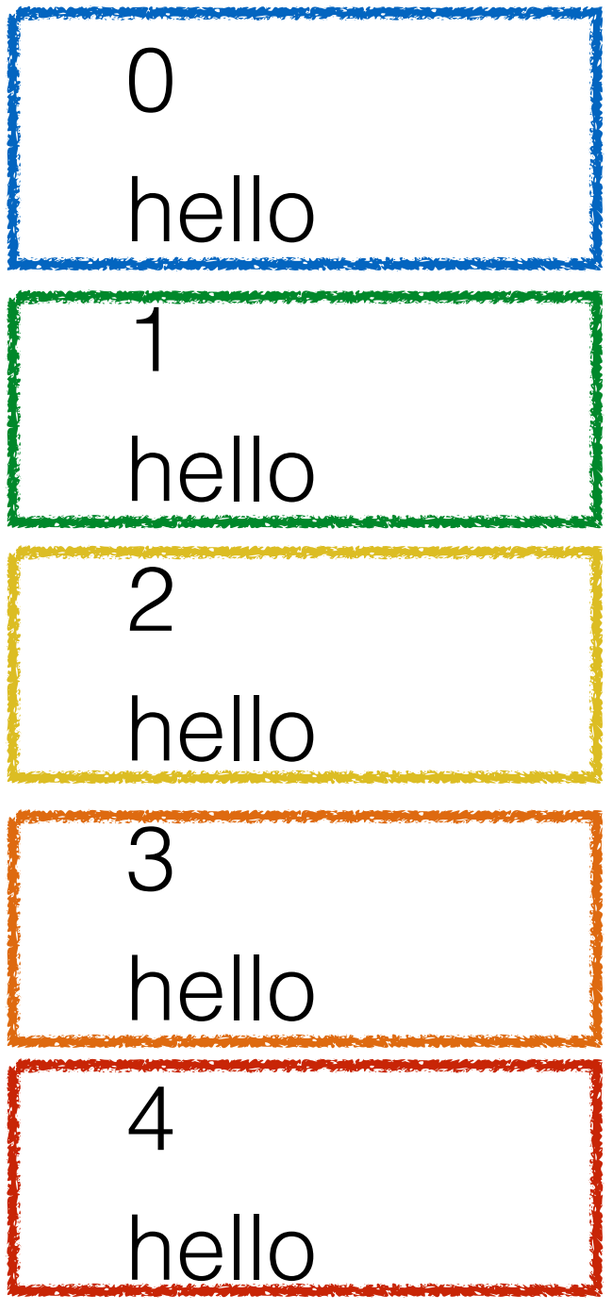
```
for counter in [0, 1, 2, 3, 4]:
    print counter
    print “hello”
```

```
0
hello
1
hello
2
hello
3
hello
4
hello
```

Repeat / Loop:
ask computer to do the dirty work

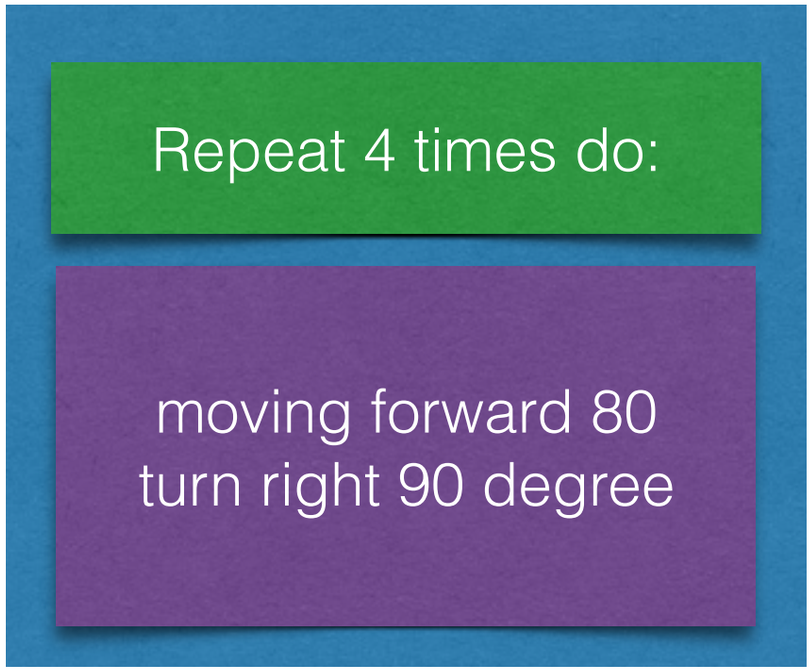
```
for counter in [0, 1, 2, 3, 4]:
    print counter
    print "hello"
```

Repeat “hello” 5 times



Repeat / Loop:
Draw a square

```
for counter in [0, 1, 2, 3]:
    yourTurtle.forward(80)
    yourTurtle.right(90)
```



Repeat / Loop:
Challenge: Draw a hexagon

```
for counter in [0, 1, 2, 3, 4, 5]:
    # your code here
```

Repeat / Loop:
Challenge: Draw a hexagon

```
for counter in [0, 1, 2, 3, 4, 5]:
    # your code here
```

```
for counter in range(6):
    # your code here
```

Repeat 6 times

Repeat / Loop:
Challenge: Draw a octagon

```
for counter in range(8):
    # your code here
```

Variables

- nick = Turtle()
- Storing Values in Variables
- Variable like a box that can hold values, you can store values inside variable with the **=** sign
- e.g. Store value 15 in a variable names spam:

```
spam = 15
print spam
```

?

https://inventwithpython.com/chapter2.html

Condition: if - else

if *a condition is evaluated to be True*:

    # do this

else:

    # do that

e.g.

```
grade = 61
if grade > 60:
    print "You passed the final exam"
else:
    print "You failed"
```

Comparison Operator

== : equal
> : greater than
< : less than
>= : greater or equal
<= : less or equal
!= : not equal

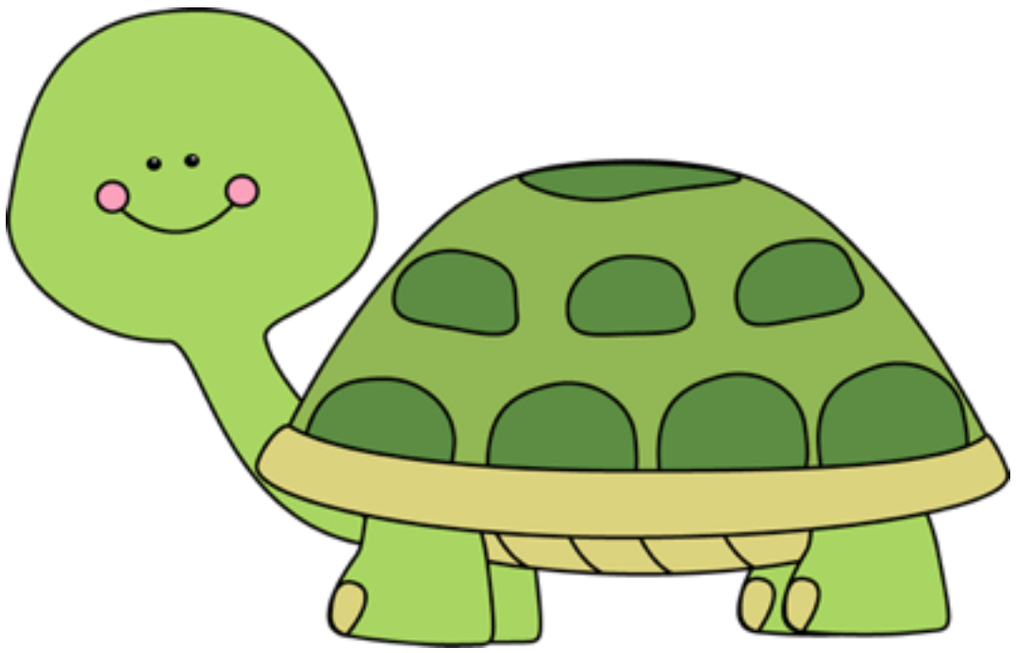
Condition: if - else cont'd

```
# Get user's input by using raw_input function
age = raw_input("What's your age?")
# age now is a string, convert it to integer
age = int(age)
# Check whether it is greater or equal than 5
if age >= 5:
    print "You can play this game"
else:
    print "You can't play this game"
```

Game Time

Help your turtle find food !

http://bit.ly/find-food



http://content.mycutegraphics.com/graphics/turtle/cute-turtle.png

Game Time

Guess the right Number !

http://ayrow.com/numbers.jpg

Recap

- **print** “hello python”
- calculator: **print** 2 * 3 + 4
- control turtle: forward / backward, right / left
- loop: **for** number **in** range(6)
- condition: **if** / **else**
- Games you made: turtle’s food, guess number

The End

Extra - Dictionary

Dictionary (key: value) pair

e.g. Check our restaurant menu:

```
menu = {
        “bacon”: 4.5,
        “sandwich”: 4,
        “waffles”: 3,
…
}
print “Price for bacon is:”, menu[“bacon”]
```

Dictionary

A waiter / waitress app

http://thumbs.dreamstime.com/z/surprised-waiter-cartoon-31089845.jpg

Challenge : Fizz Buzz
A real interview question for CS student

- Problem: from 1 to 20,
    - if the number is multiple of 3, print “fizz”;
    - if the number is multiple of 5, print “buzz”;
    - if the number is both multiple of 3 and 5, print “fizz buzz”;
    - otherwise, print out the number

The Real End

Q & A