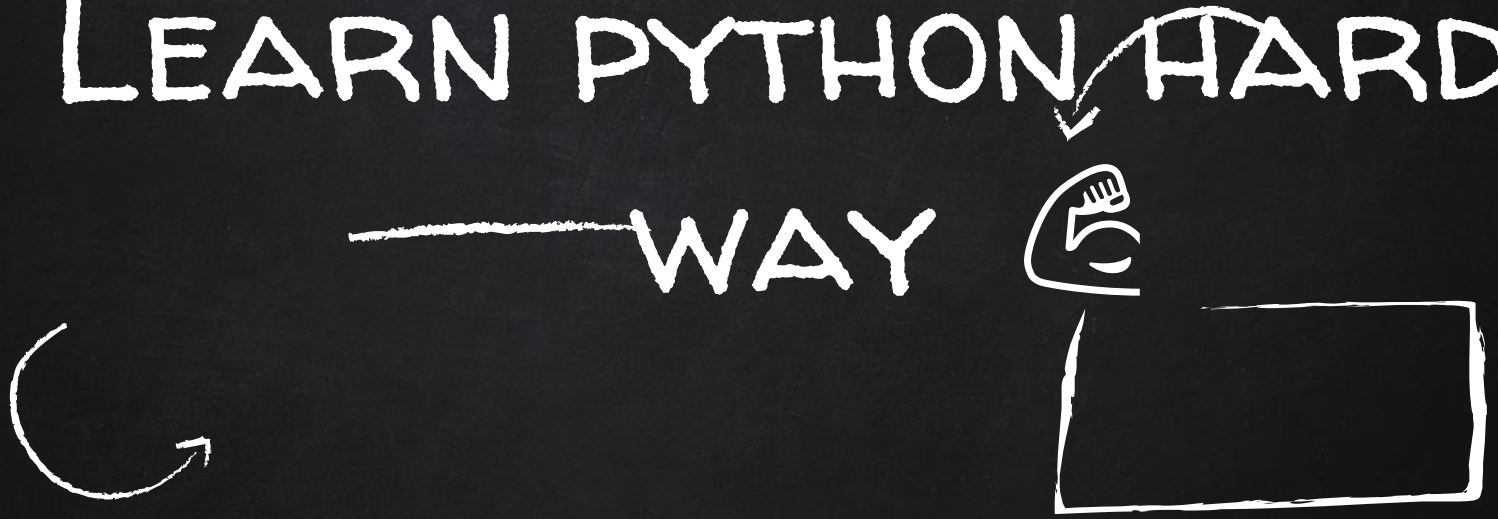
Learn python hard way



Today's lesson

- By the end of this lecture you should be able to:
  -

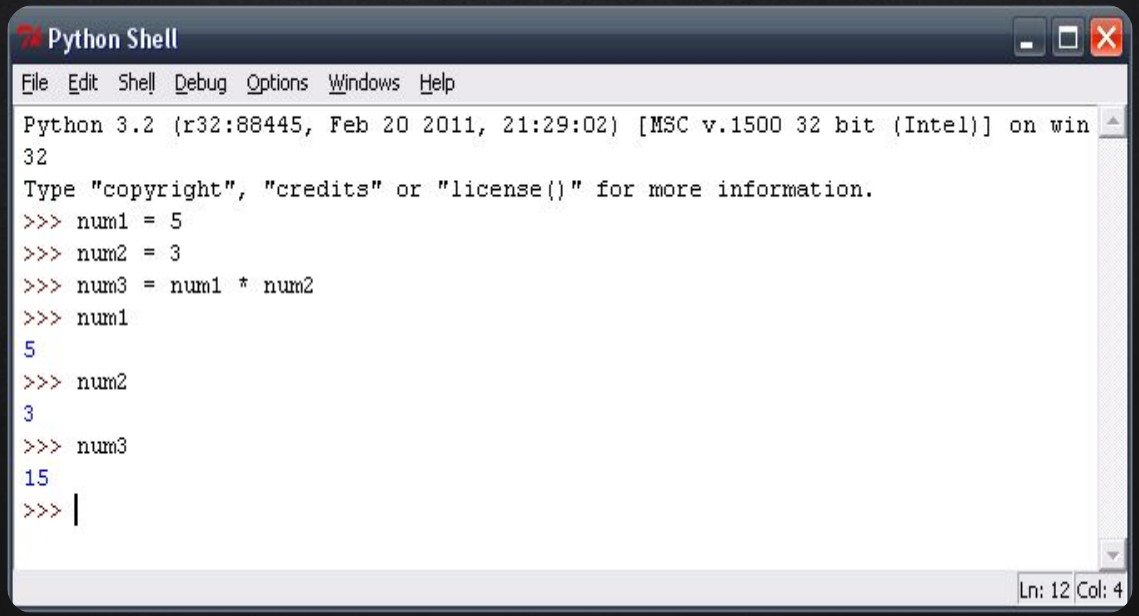
Terminology Review

- Statement
  - One programming instruction.
    - Assignment statement, Return statement, Conditional statement
- Variable
  - used to store variables.
- Comment
  - ignored by the compiler and interpreter
- Data Types
- Operators
  - symbols that define certain operations (addition, division, remainder etc etc)

Operators

- Each programming language has a set of operators that define certain operations. Common types of operators such as
  - Assignment Operator
  - Arithmetic Operators
  - Bitwise Operators
  - Logical Operators
  - String Operators
  - Comparison Operators

ASSIGNMENT OPERATOR



Arithmetic operator

Operator	Meaning	Example
+	Addition	4 + 7 ⟶ 11
-	Subtraction	12 - 5 ⟶ 7
*	Multiplication	6 * 6 ⟶ 36
/	Division	30 / 5 ⟶ 6
%	Modulus	10 % 4 ⟶ 2
//	Quotient	18 // 5 ⟶ 3
**	Exponent	3 ** 5 ⟶ 243

Comparison operators

Operator	Meaning
=	Has a value of
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
< >	Not equal to

Logical Operators

TRUTH TABLES OF LOGICAL OPERATIOS

AND			OR			NOT	
X	Y	X.Y	X	Y	X+Y	X	X'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

OPERATORS SHORTHANDS

Shorthand operator	Meaning
x += y	x = x + y
x -= y	x = x – y
x *= y	x = x * y
x /= y	x = x / y
x %= y	x = x % y
x <<= y	x = x << y
x >>= y	x = x >> y
x >>>= y	x = x >>> y
x &= y	x = x & y
x ^= y	x = x ^ y
x |= y	x = x | y

Boolean logic practice

EVALUATE THE FOLLOWING STATEMENTS

1. 1 == 1 and 1 == 1
2. 1 == 1 and 1 == 0
3. "test" != "testing"
4. not (True and False)
5. True and 1 == 1

CLASS ACTIVITY TO SUBMIT

if student gets 90 or higher to 100: they get an A
If students get 80 or above and less than 90: they get a B
If students get 70 or more and less than 80: they get a C
If students get 55 or above: and less than 70 they get a D
Any grade lower than 55 is F

Loops

are used to repeat a statement or a block of code for number of times

Types of loops

- For Loop:
  - repeats a statement for a number of time
- While loops:
  - runs for as long as a condition is true

For Loop

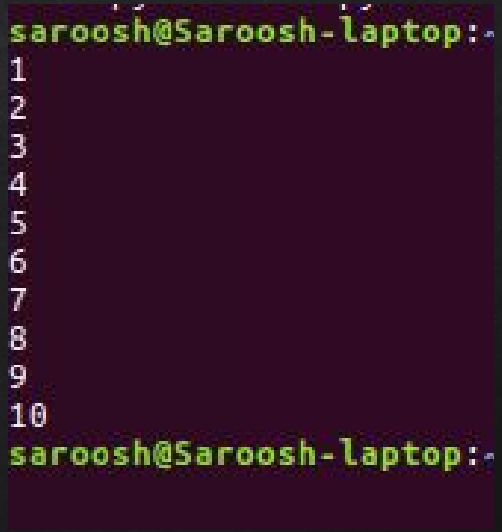
- for loop in python uses a range function to run

```
for x in range(0, 3):
    print "We're on time %d" % (x)
```

FOR LOOP EXAMPLE

- print 10 lines which use the same code

```
 4  print(1)
 5  print(2)
 6  print(3)
 7  print(4)
 8  print(5)
 9  print(6)
10  print(7)
11  print(8)
12  print(9)
13  print(10)
14
```



FOR LOOP

- ✘ replacing 10 lines with 2 lines, a loop

```
5  for i in range(0, 10):
6      print(i)
7
```

```
saroosh@Saroosh-laptop:~/
0
1
2
3
4
5
6
7
8
9
saroosh@Saroosh-laptop:~/
```

RANGE FUNCTION

```
5   for i in range(10):
6       print(i)
7
```

```
saroosh@Saroosh-laptop:~/Documents/Worl
0
1
2
3
4
5
6
7
8
9
saroosh@Saroosh-laptop:~/Documents/Worl
```

RANGE FUNCTION

```
13  for i in range(0, 10, 2):
14      print(i)
15
```

```
saroosh@Saroosh-laptop:~/Documents/W
0
2
4
6
8
saroosh@Saroosh-laptop:~/Documents/W
```

In-class exercises

1. Write a for loop that prints the numbers from 5 to 15
2. Write a for loop that prints the numbers from 5 to 15 in increments of 3
3. print multiples of 2 from 12 to 30
4. print multiples of 3 from 3 to 90
5. a loop that prints out letters of the word 'banana'

While loop

- ✘ a while loop runs a code as long as a condition is true
- ✘ for example: print numbers as long as they are less than 5

```
i = 0
print('running while loop')
while (i < 5):
    print(i)
    i = i + 1
```

```
running while loop
0
1
2
3
4
saroosh@Saroosh-laptop:~/Documents/W
```

LISTS

a data structure to store data

LIST

```
22  my_list = [1,2,3,4,5]
23  print(my_list)
```

```
8
running while loop
0
1
2
3
4
[1, 2, 3, 4, 5]
saroosh@Saroosh-laptop:~/Documents/Work/PerScholas/python/AlumniPythonClass$
```