

IDC PROGRAMMING COURSE BY
BIJOY AND DHEERENDRA

PYTHON 3

**INSTALL
PYTHON**

LINUX

- ▶ Confirm you need to install - open terminal and type `python3`
- ▶ If `python3`'s console doesn't open type - update your OS, Linux comes with Python 3!
- ▶ Don't want to update? - `sudo apt-get install python3`

WINDOWS AND MAC

- ▶ Install from python's webpage in the download section. Installation is straightforward.

**INSTALL
PYCHARM**

INSTALL PYCHARM

- ▶ Google PyCharm or go to <https://www.jetbrains.com/pycharm/>
- ▶ Download the Community version for your Operating Systems
- ▶ Install based on your OS
- ▶ Any Trouble? Ask Us

HELLO WORLD!

GETTING STARTED

- ▶ Python is a simple language yet very powerful
- ▶ We will begin by simply printing things on the screen

```
print('Hello, World!')  
  
print('Your Name')  
  
print(YOUR_AGE)  
  
print('IDC, IIT Bombay')
```


PROGRAMMING BASICS

WHAT ARE DATA TYPES?

- ▶ **Integers** are like 1, 0, -1, 21, 232,
- ▶ **Floating Point** are like 0.1, 0.6, 1.3e-2, 9.3233, ...
- ▶ **String** are like 'Hello, World!', 'Yo!', ...
- ▶ **Boolean** are True, False

VARIABLES

- ▶ These are simply names given to values that you want to use later.

```
name = 'Your Name'
```

```
print(name)
```

```
age = YOUR_AGE
```

```
print(age)
```

PLAYING AROUND WITH VARIABLES AND NUMBERS

- ▶ 1 dozen Banana (Rs 20 p.d.) and 3 Oranges (Rs 65 p.d.).
Whats the total cost?

```
cost_of_bananas = 1 * 20
```

```
cost_of_oranges = (3 / 12) * 65
```

```
total_cost = cost_of_bananas + cost_of_oranges
```

```
print(total_cost)
```



WE ARE BACK

WHAT DO YOU THINK IS THE OUTPUT

```
a = b  
  
b = 10  
  
print (a, b)
```

```
a = 'Hello'  
  
a = 'World'  
  
print(a)
```

```
a = b = 10  
  
print (a, b)  
  
a = 20  
  
print (a, b)
```

PLAYING AROUND WITH STRINGS

- ▶ Let's try and play around with strings

```
first_name = 'Giorgio'

last_name = 'Armani'

name = first_name + ' ' + last_name

print(name)
```

- ▶ Now try doing this

```
something = 'Hello' + 31
```

DATA TYPE CASTING

- ▶ Often you wish to convert one datatype into another.
- ▶ Say you have a number as a string, say '21', and you want to add 12 to this number.

```
number_string = '21'  
  
number = int(number_string)  
  
number = number + 12  
  
number_string = str(number)
```


PLAYING AROUND WITH STRINGS

- ▶ Indexing in most programming languages starts from 0
- ▶ Getting the n^{th} character of a string is simply `string[n - 1]`

```
name = 'Giorgio'

print(name[2])

print(name[2:5])

print(name[2:])

print(name[:5])
```

READING VARIABLES FROM THE USER

- ▶ Now we will learn how to read variables from the user

```
name = input('Enter Your Name')  
  
age = int(input('Enter Your Age'))  
  
height = float(input('Enter Your Height'))
```

TRY IT YOURSELF

```
name = input('Hi, please enter your name')  
  
print('Hello', name)  
  
college = input('Where do study?')  
  
print('It must be awesome to be a student at', college)
```

- ▶ Now add more stuff, like ask the users age!
- ▶ We will soon look into how you can actually do stuff with the values you get!

PYTHON

BASICS

FUNCTIONS

REVISITING THE FRUIT SHOP

- ▶ You are really fond of bananas and oranges apparently, and keep coming back to buy them again and again!
- ▶ Now, you know Bananas are Rs 20 a dozen, and oranges Rs 65 a dozen.
- ▶ We tried writing this last time, but now instead of 1 dozen bananas and 3 oranges, I take 11 bananas, and 15 oranges. Whats the total? What if I take 15 bananas and 11 oranges?

USING FUNCTIONS TO MAKE LIFE EASIER!

```
def get_total_cost(number_of_bananas, number_of_oranges):  
    cost_of_bananas = (number_of_bananas / 12) * 20  
    cost_of_oranges = (number_of_oranges / 12) * 65  
    total_cost = cost_of_bananas + cost_of_oranges  
    return total_cost  
  
cost = get_total_cost(11, 15)  
  
print(cost)
```

TRY IT YOURSELF

- ▶ Take in the input from the user and,
- ▶ Make a function which calculates the total cost, and allows you to even change the price of the banana and orange.
- ▶ (Later, try to figure out from Google, how to add default arguments to a function)

WHAT DO YOU THINK IS THE OUTPUT

```
a = 10
```

```
b = 20
```

```
def f (a):
```

```
    b = a + 10
```

```
    return b
```

```
c = f (5)
```

```
print(a, b, c)
```


REVISING WHAT WE LEARNT

1. Write a function which takes
2. in user's name, then prints
3. out the first 5 characters of
4. the name.
5. Write a function which returns the
6. cube of a number

PROGRAMMING

BASICS

DECISION MAKING

A SIMPLE SCENARIO : THE **IF** TAG

- ▶ You wish to buy a car. But only if the price of the car is less than 10,00,000 units, you buy a car.
- ▶ The code is really simple! Note that **spacing is important!**

```
if car_price < 1000000:  
    print('Buy a Car')
```

A SIMPLE SCENARIO : THE **ELSE** TAG

- ▶ You wish to buy a car. This time if the price of the car is not less than 10,00,000 units, you buy a bike instead

```
if car_price < 1000000:  
    print('Buy a Car')  
  
else:  
    print('Buy a Bike')
```

A SIMPLE SCENARIO : THE **ELIF** TAG

- ▶ You wish to buy a car. This time if the price of the car is not less than 10,00,000 units, you buy a bike only if its price less than 50,000.

```
if car_price < 1000000:  
    print('Buy a Car')  
  
elif bike_price < 50000:  
    print('Buy a Bike')
```

TRY IT YOURSELF

- ▶ You wish to buy a car. You will buy it only if the car is cheaper than 12,00,000. Else you go to think about buying a bike. You'll buy the bike only if the bike is more expensive than 60,000. Otherwise you simply buy a bicycle.
- ▶ Try writing the code for this!

PROGRAMMING

BASICS

LOGICAL OPERATORS

SIMPLE OPERATORS

- ▶ Some operators are really simple and intuitive

LESS THAN

<

LESS THAN EQUAL

<=

MORE THAN

>

MORE THAN EQUAL

>=

SIMPLE EXAMPLES

```
age < 21
```

```
height >= 180
```

```
weight <= 65
```

```
roll_number > 50
```

```
# Can even have variables
```

```
my_age <= your_age
```

```
my_weight > your_weight
```

SIMPLE OPERATORS

- ▶ Some operators are really simple and intuitive

EQUAL

==

NOT EQUAL

!=

AND

and

OR

or

SIMPLE EXAMPLES

```
age == 21
```

```
height != 180
```

```
age > 20 and height < 180
```

```
age <= 21 or weight < 65
```

SAMPLE CODE

```
if name[0] == 'a' or name[0] == 'A':  
    print('Your name starts with a A')  
  
elif name[0] == 'B' and name[1] != 'i':  
    print('Your name starts with a B, but the second is not i')  
  
else:  
    print('Your name is ' + name)
```

TRY IT YOURSELF

- ▶ Given a name, print 'Hi' if it's length is between 2 and 10. Else if the name is 'Adele' print 'Hello', or if it starts with 'Ada' then print 'Yo'. Else print 'Hey'.
- ▶ Hint: To find the length of the string use `len(your_string)`
- ▶ Hint: Experiment with `your_string[0:3]`

PROGRAMMING

BASICS

LOOPS

A SIMPLE SCENARIO : THE **WHILE** TAG

- ▶ Print all numbers from 1 to 10
- ▶ We will use a while loop, which iterates while a condition is satisfied

```
counter = 1  
  
while counter <= 10:  
    print(counter)  
    counter = counter + 1
```

A SIMPLE SCENARIO : THE **FOR** TAG

- ▶ Print all numbers from 1 to 10
- ▶ We will use a for loop, which iterates in a series or a list of elements

```
# Note that the last argument of range is not reached  
for counter in range(1, 11):  
    print(counter)
```


TRY IT YOURSELF

Using only a while loop, print numbers from 1 - 20 only EVEN numbers

Using only a for loop, print numbers from 1 - 20 only EVEN numbers

Famous Introductory problem to Loops:

Take a user input, if it is 42 print "Answer to life universe and everything". Else keep taking the input till this happens

TRY IT YOURSELF

- ▶ Write a function which prints 'n' rows like these

*

* *

* * *

* * * *

* * * * *

.....

* * * * *
..... *

TRY IT YOURSELF

- ▶ Given a word, print out it's characters once in every line.
- ▶ So `printCharacters("Hello")` should print H,e,l,l,o in separate lines each



PYTHON

BASICS

LISTS

WHAT IS A LIST

- ▶ It is simply a sequenced collection of objects.
- ▶ For people who came from C++/Java, they can think of lists as very similar to vectors or arrays.
- ▶ Usage is very simple a list is simple a comma separated list of elements in block-brackets.
- ▶ `example_list = [1, 3, 'Hi', 4, True, 0.42, 2]`

COMMON LIST OPERATIONS

- ▶ `your_list.append(a)`
- ▶ `your_list.pop(i)`
- ▶ `your_list.remove(a)`
- ▶ `your_list.insert(i, a)`
- ▶ `your_list[3], your_list[-2]`
- ▶ `your_list[2:5], your_list[:6], your_list[-3:]`

SEE IT IN ACTION

```
a = [1, 2, 3]

b = []

for position in range(0, len(a)):

    b.append(a[position])

    b.insert(0, a[position])
```

```
a = [1, 2, 3]

b = []

for item in a:

    b.append(item)

    b.insert(0, item)
```


TRY IT YOURSELF

Write a function which takes 2 arguments start, stop. It will return a list of all even numbers between start and stop.

Write a function which takes a list as argument and removes the first even number in it. Using a) Remove b) Pop

(Challenge) Write a function which removes all the even numbers

(Easy way out, but bad, keep calling the above function, but don't do that :P)

Write a function which takes a list, start_value and stop_value as arguments and prints elements from start_value and stop_value