



Python Introduction

Let's dive in!



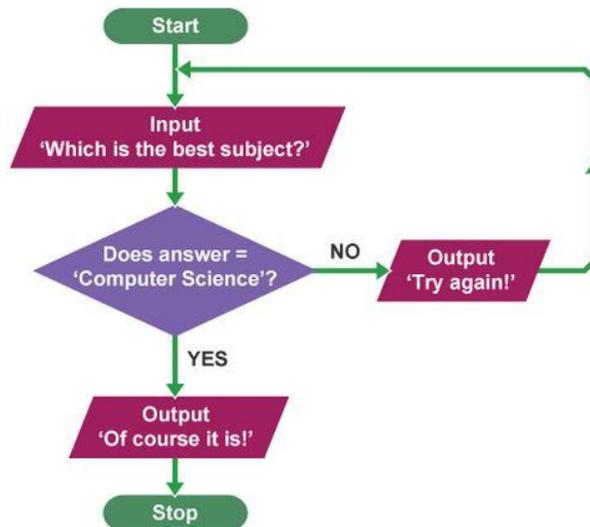
Remember computer programming is about **solving problems**. We can solve tons of cool problems using Python. You will eventually learn to create really neat tools and fun applications using Python.

But first we have to **learn some basics**. We need to get familiar with some key principles before we can use Python effectively. The next few lessons are designed to get you familiar with the basics of Python.

All computer programming languages, really, have a few key aspects:

1. **Input**
2. **Output**
3. **Decisions**
4. **Calculations**
5. **Repetition**

Python will give you tools in **all** of these areas. Let's take a look!

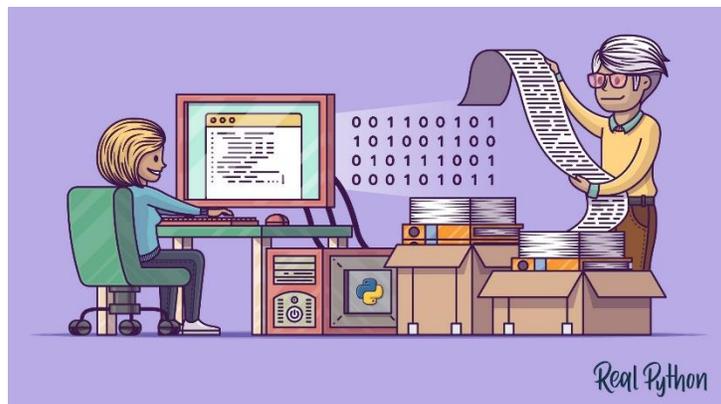


Input/Output/Variables:



Output:

Generally, Python can **output** (produce): lists, numbers, words, and images. The most interesting and easiest are outputs are **words** and **images** so let's start by practicing creating these simple **outputs**:



Exercise#1

Type the following into trinket (don't cut and paste! – Type it in please)

```
print("I am a cool student.")  
print("I am learning Python.
```

Type the following into trinket (don't cut and paste! – Type it in please)

```
print("Im the best student in the world!")  
print("\n")  
print("I am learning Python.")
```

make sure you know what “\n” does?

Type the following into trinket

```
print("Im the best \r student in the world!")  
print("I am learning Python.")
```

What doesn /r do?

/r and /n are called: **Escape Characters**

Escape Characters

Escape Characters are special characters that help us organize written output. There are many of these. Here is a short list of some you

Escape character	Prints as
\'	Single quote
\"	Double quote
\t	Tab
\n	Newline (line break)
\\	Backslash
\r	Return Carriage similar to Newline

Now you can get words to appear on a screen using Python. But that's not *too* exciting... you can do that by just typing into a word doc! Let's keep going:

Too Easy!

Input/Variables:



One way for Python to gather information is by getting it from **humans**. Humans can type words, numbers, lists, and commands into python. We can then store these items in **variables**.

A **variable** (in all computer programming languages) is: a way to label and store information...kinda like putting something into a labelled box. Let's have a look to see how this works:

Exercise#2

Type the following into trinket

```
age=17
name='Jeff'
fav_food='apples'

print(age)
print(name)
print(fav_food)

#change my favorite food :)

fav_food='banana'
print(fav_food)

print("\n")           # leave a space
fav_food='toast'      # changed my favorite food again!
print('my name is')
print(name)
print('my favorite food is')
print(fav_food)
```

When you see a hashtag # in python, this means you have created a "comment" in your code. Comments are ignored by the computer, but are handy for humans who like to put comments and reminders in their code



See how we are **creating storage spaces** and then **stuffing things into them**? We do this with the = (equals sign)...which in programming does **NOT** mean "equal to"...it means "*assign value of*"

Type the following in to trinket!...more fun examples with **Variables**:

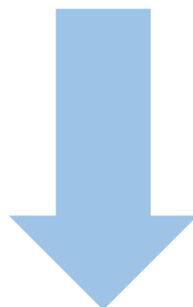
```
team_players=23
extras=5
total= team_players + extras
print('The number of regular players on our team is')
print(team_players)
print('We have this may extra players')
print(extras)
print ('Total amount of players is')
print(total)
print('\r')
extras=10
total= team_players + extras
print ('Now we have his many extas!')
print(extras)
print('So our total number of players is')
print(total)
```

Now you can:

1. print stuff to the screen and
2. use **variables!**

Good for you!

Later we will look at how to **draw images** with python, but for now let's look at **input**:



Input



Collecting information is a key part of computer programming. One way python does this is with the `input()` command.

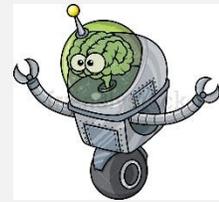
The `input()` command allows python to **accept typed info from a human**.



Exercise#3

Enter the following code into trinket. When you run the program. The `input()` command, the computer will prompt you to enter information. Try it out.

```
print ('Hi, My name is Calcutron, Im from planet Zerp')
name=input('what is your name?')
print ('Hi',name)
planet=input('What is the name of your planet?')
print(planet,'?????')
print('Doesnt sound too cool. Im naming your planet Salad Ball!')
print('You live on Salad Ball!')
print('You like that',name, '?')
answer=input('yes or no?')
if answer=='yes':
    print('Good!',name,'Who is your King on planet Salad Ball?')
if answer=='no':
    print(name,'!!!! dont be such a poor sport!')
    print('Salad Ball!'*3)
```



Notice the “**if**” statements above...we will talk about later but it’s not to hard to see what and ‘**if**” statement does from the example. **INDENTING** – notice the **indenting** – very very very important!

Exercise#4 - your turn!

Create a new program similar to the one above (you can use the code above as a template). Your goal will be to create a character that someone can have a dialog with. Make is as **fun and as interesting as possible** for the user!....show Mr. Walzl when you are done. You will be marked on this!

Need a creative boost? Maybe use the scenario of a person going to buy something from the corner store.

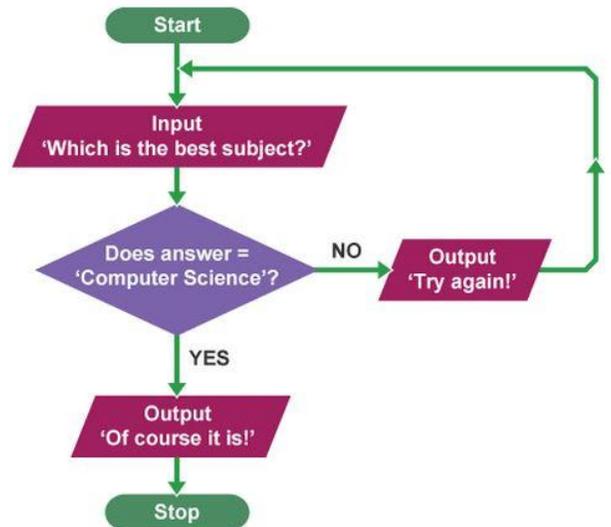


REMBMER!

Computer programming is about:

1. Input
2. Output
3. Decisions
4. Calculations
5. Repetition

We have looked at output, **input**, and peeked at **decisions**. Let's get some practice with **Calculations** and **Repetition** using Python.

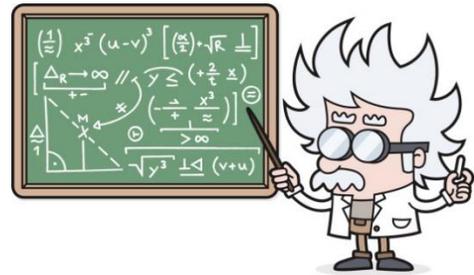


Calculations:

An important aspect of computer programming is that you can **easily** create a procedure for doing difficult calculations. And then use this procedure over and over again without much effort.

It is easy to use Python to make calculations.

Here is a list of some of the character you need to know to code calculations in Python:



Numbers and math

Operator	Description
+ plus	Sum
- minus	Subtraction
/ slash	Floor division
* asterisk	Multiplication
** double asterisk	Exponentiation
% percent	Remainder
< less-than	Comparison
> greater-than	Comparison
<= less-than-equal	Comparison
>= greater-than-equal	Comparison

Exercise#5

Type the following into trinket



```
print('lets calculate the area of a rectangle')
length=int(input('Give me the Length of your rectangle'))
width=int(input('Gove me the width of your rectangle'))
area=length*width
print 'the length of your rectangle is:',area,'cm squared'
```

Exercise#6

Your turn! - **create 6 mini programs** like the one above.

Each program should be use a different mathematical operation form the list below. Show

Mr. Walzl when you are done. This is for marks.

Make sure you understand what each of the symbols below does.

NOTE:

int - means we have to change the input from a **string** (list of characters) into an **integer** (a whole number).

Operator	Description
+ plus	Sum
- minus	Subtraction
/ slash	Floor division
* asterisk	Multiplication
** double asterisk	Exponentiation
% percent	Remainder

