Launchpad Learning

Empowering students with the skills required to lead the way in hi-tech world.

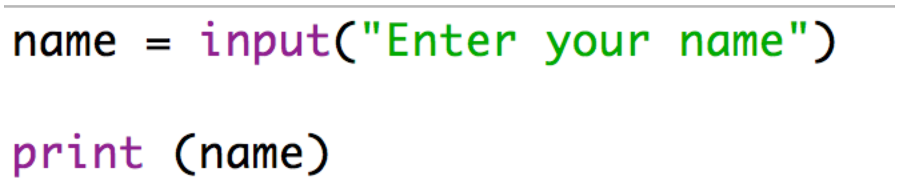
Launchpad Learning

Section 2
Conditionals & Control Flow

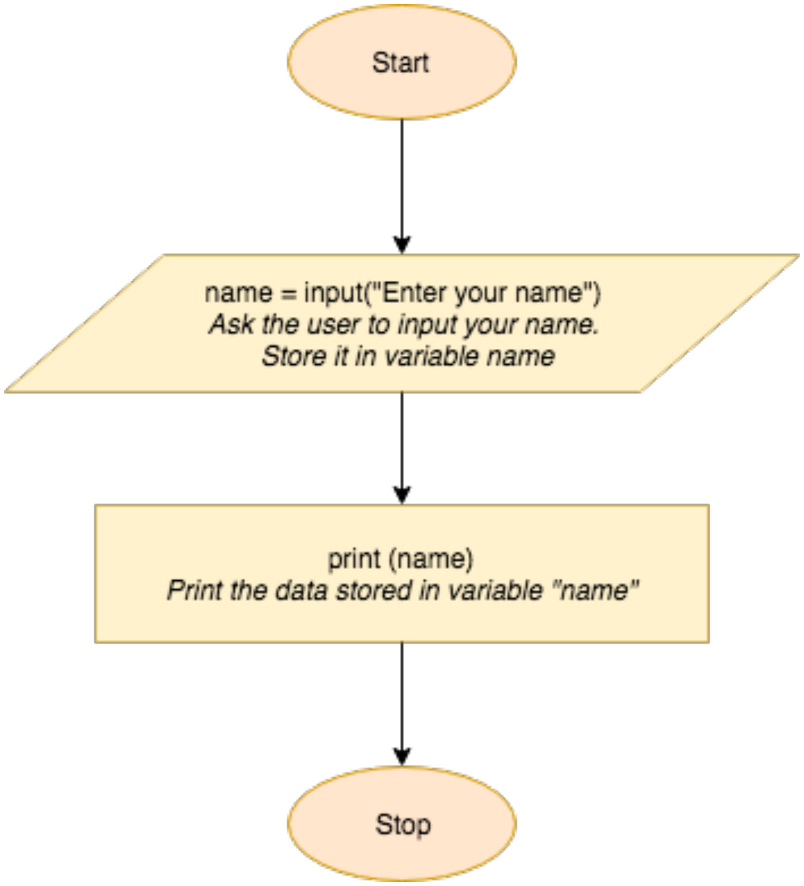
Recap

- Variables – placeholders for your data
- String – sequence of characters (letters, numbers, symbols etc.)
- Keywords like print, input etc. give instructions to Python interpreter to perform some functions
- Python interpreter reads each line of your program and runs it in the order it reads it

Program Execution



```
name = input("Enter your name")

print (name)
```



Start

name = input("Enter your name")
Ask the user to input your name.
Store it in variable name

print (name)
Print the data stored in variable "name"

Stop

Note that the program reads each statement one by one and executes in the same order.

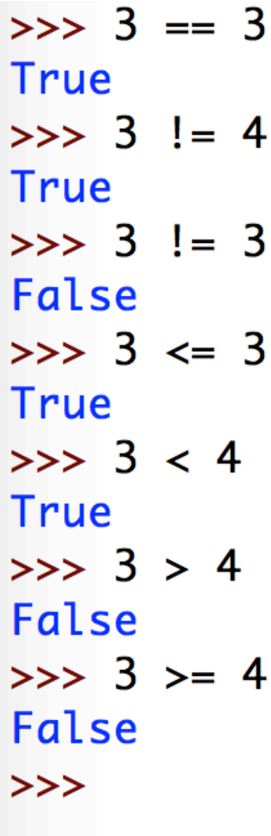
This is all good for simple programs but most programs are not this simple

We use what we call conditionals and control flow to make the program skip some statements or repeat some statements

Comparators

- Building blocks for conditionals, used to compare two data values

- Different types:
  - Equal to (==)
  - Not equal to (!=)
  - Greater than (>)
  - Greater than equal to (>=)
  - Less than (<)
  - Less than equal to (<=)



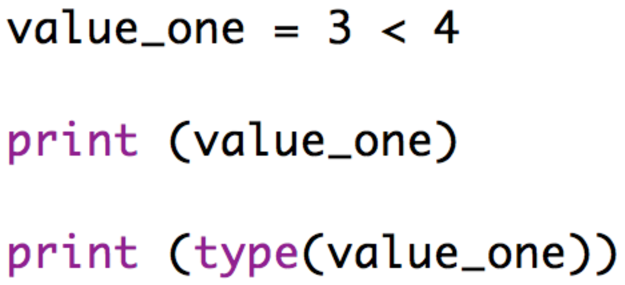
```
>>> 3 == 3
True
>>> 3 != 4
True
>>> 3 != 3
False
>>> 3 <= 3
True
>>> 3 < 4
True
>>> 3 > 4
False
>>> 3 >= 4
False
>>>
```

Exercise

Expression	Value
90 == (2 * 45)	True
-25 > -10	
10 != (5 + 5)	
5 >= 5	
6 <= 100	
17 < 328	
50 > 199	
10 > 3**2	

Boolean

- Boolean data type takes just two values – True/False
- Variables can store Boolean values
- Try the following code in IDLE:



```
value_one = 3 < 4

print (value_one)

print (type(value_one))
```

Boolean Operators

- Three major Boolean operators:
  - AND (and, &)
  - OR (or, |)
  - NOT (not, !)

NOT	
A	not A
True	False
False	True

- Boolean Operators return Boolean values (True/False)

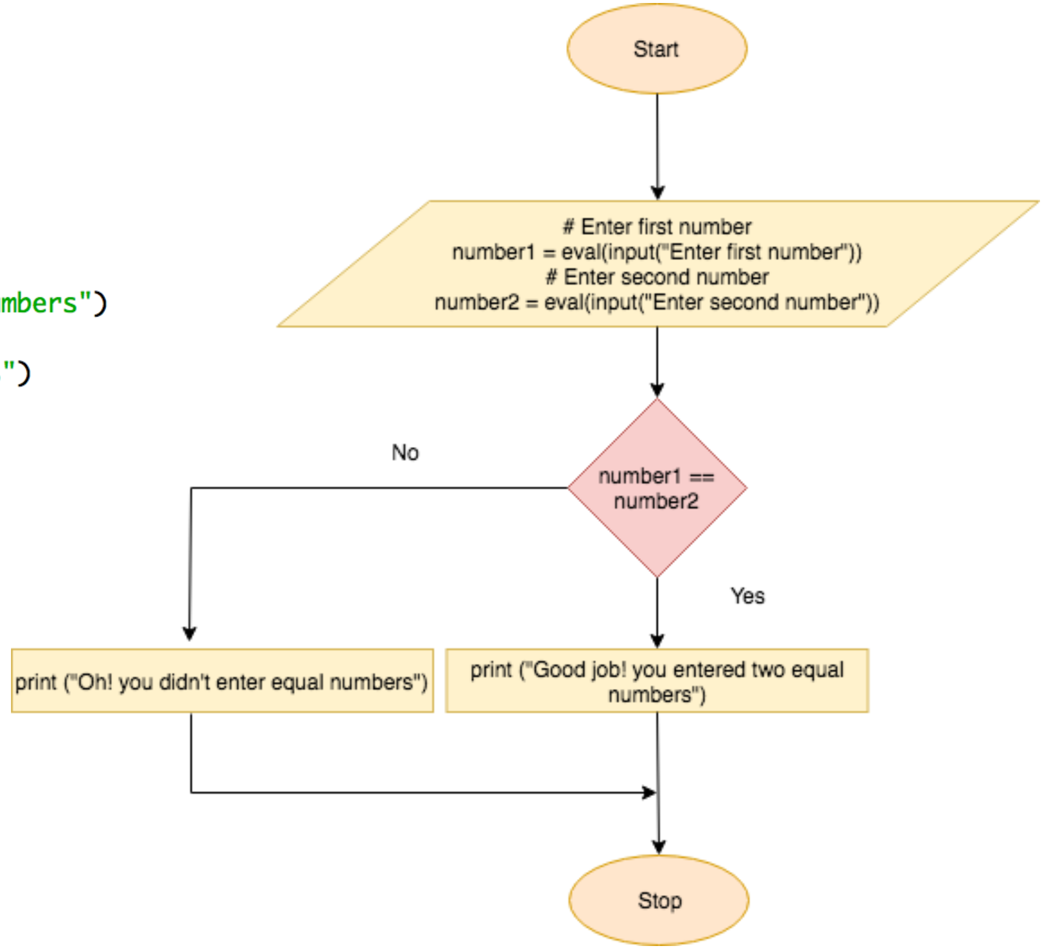
AND			OR		
A	B	A and B	A	B	A or B
True	True	True	True	True	True
True	False	False	True	False	True
False	True	False	False	True	True
False	False	False	False	False	False

Exercise

Expression	Value
(3 > 4) and (4 > 3)	False
not (3 == 3)	
(4 >= 5) & (3 < 4)	
(2 == 2) or (2 != 4)	
(2 > 2) | (3 > 4)	
!(3 > 4)	
not not False	

Conditional Statement

```
# Enter first number
number1 = eval(input("Enter first number "))

# Enter second number
number2 = eval(input("Enter second number "))

# Check if both numbers are same
if number1 == number2:
    print ("Good job! you entered two equal numbers")
else:
    print ("Oh! you didn't enter equal numbers")
```



Start

# Enter first number
number1 = eval(input("Enter first number"))
# Enter second number
number2 = eval(input("Enter second number"))

number1 == number2

No

Yes

print ("Oh! you didn't enter equal numbers")

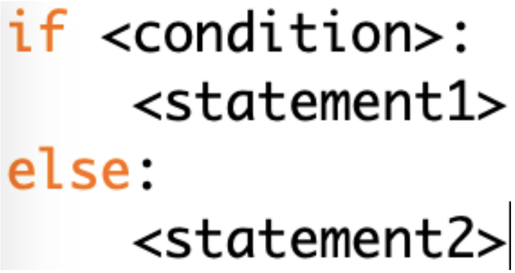
print ("Good job! you entered two equal numbers")

Stop

If/Else

- As the name suggests, If and Else statements are used to check if the condition holds

- Syntax



```
if <condition>:
    <statement1>
else:
    <statement2>
```

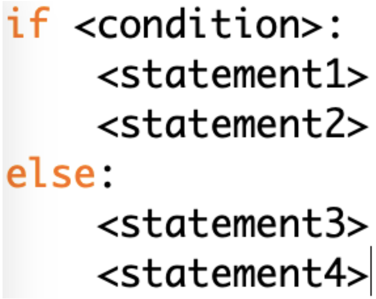
- Notice that the if statement has a keyword “if” which is followed by a Boolean condition that should evaluate to either True/False

Example 1

- Write code that checks if the number entered by the user is positive or not ?
- Step 1: Input the number
- Step 2: Write the if loop (if num > 0)
- Step 3: Complete the else part

Compounded If Statements

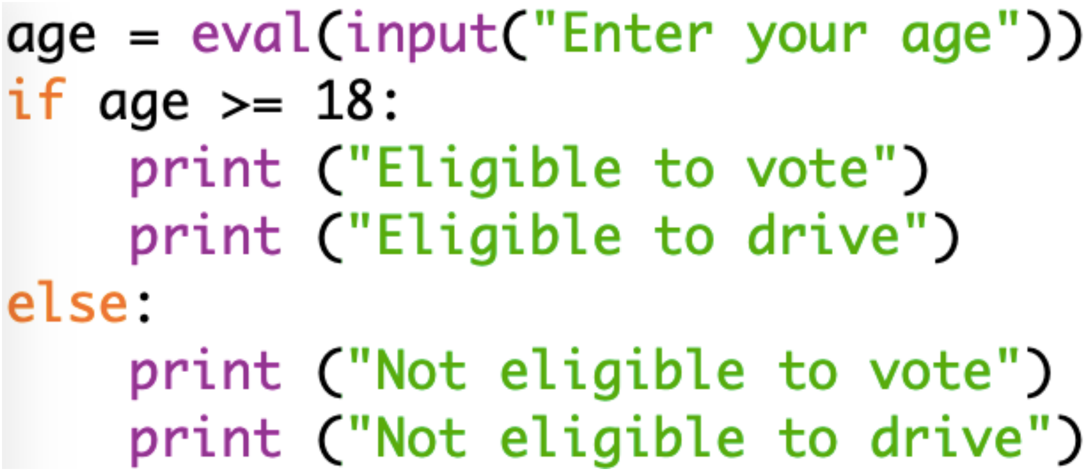
- You can add multiple statements in the if condition:



```
if <condition>:
    <statement1>
    <statement2>
else:
    <statement3>
    <statement4>
```

- Write code for the following condition:
  - If age of person is more than 18
    - Then person is eligible to vote
    - Then person is eligible to drive

Solution



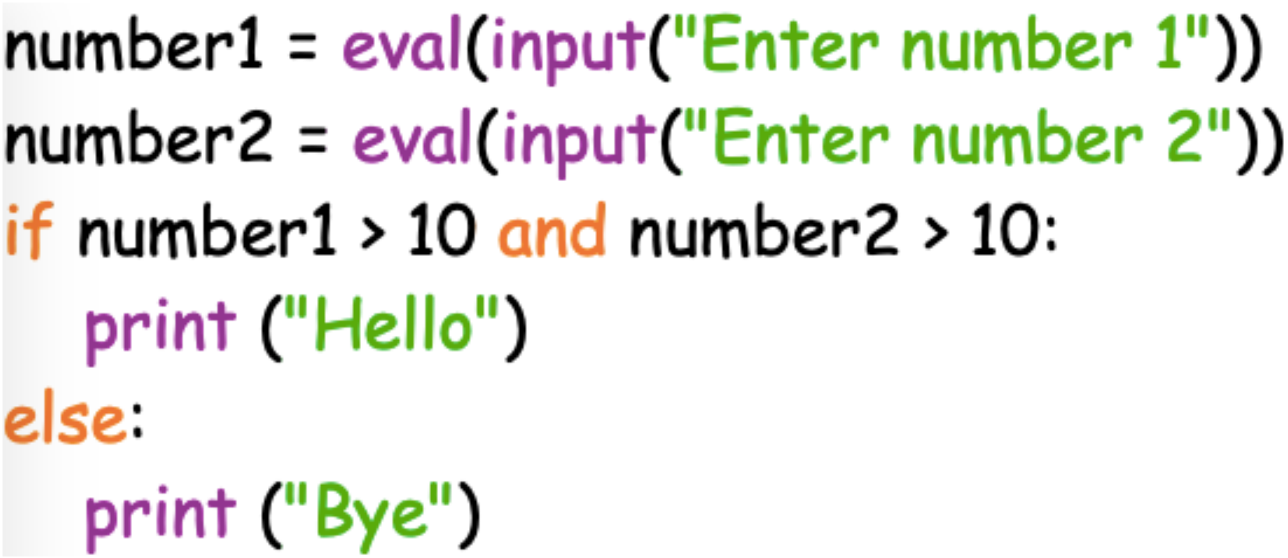
```python
age = eval(input("Enter your age"))
if age >= 18:
    print ("Eligible to vote")
    print ("Eligible to drive")
else:
    print ("Not eligible to vote")
    print ("Not eligible to drive")
```

To get the solution please download test1.py from the portal. Once you have the solution, change if condition such that a person who is 15 years or above is eligible to vote and drive

Question

- Write a program that takes in two numbers as input from user and store it in variables number1 and number2
  - Print ”hello” if both the numbers are greater than 10
  - Otherwise, print ”bye”

Solution



```
number1 = eval(input("Enter number 1"))
number2 = eval(input("Enter number 2"))
if number1 > 10 and number2 > 10:
    print ("Hello")
else:
    print ("Bye")
```

To get the solution please download test2.py from the portal. Once you have the solution, change if condition such that if number1 is greater than 15 and number2 is smaller than 15, print (“Hello”) otherwise print (“Bye”)

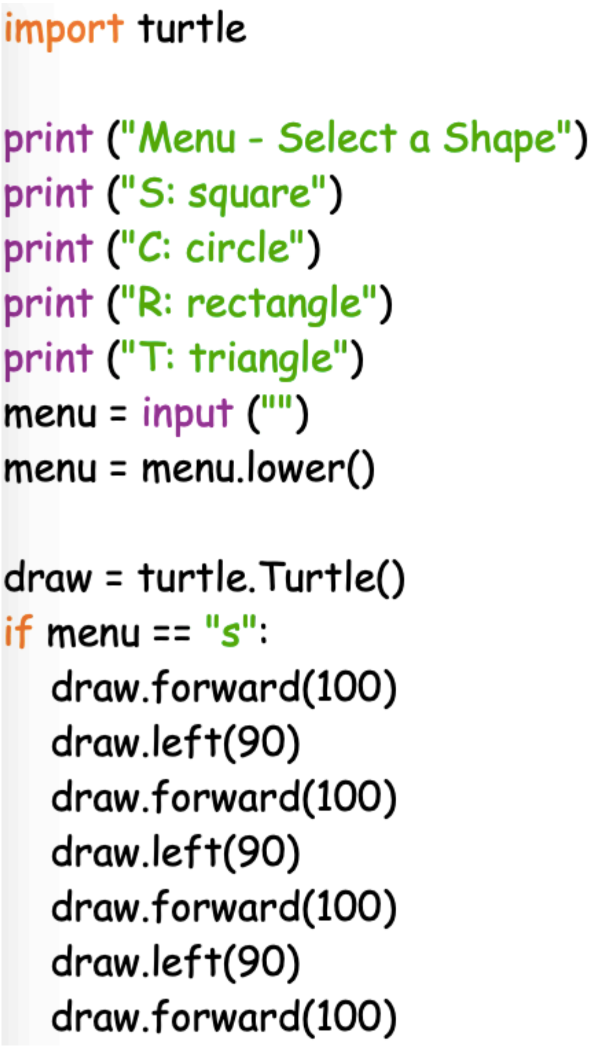
Mini Turtle Game

- Download Shapes.py from the portal

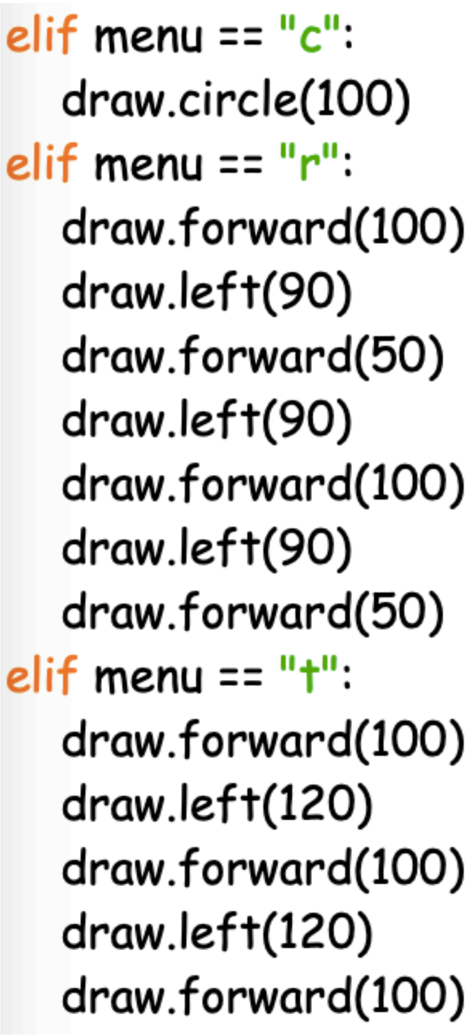
- Understand the code:
  - Hint: Turtle is a graphic library which is used to paint and draw lines and shapes
  - draw is a variable of type turtle, forward(100) means move forward by 100 steps
  - left(90) means turn left

- Task:
  - You can see that we are able to create different shapes like square and circle. Complete the code to create rectangle and then add triangle to the code as well. Solution is in ShapesSolution.py

Solution



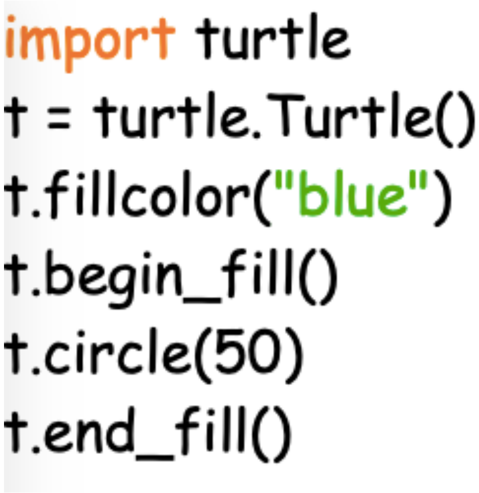
```
import turtle

print ("Menu - Select a Shape")
print ("S: square")
print ("C: circle")
print ("R: rectangle")
print ("T: triangle")
menu = input ("")
menu = menu.lower()

draw = turtle.Turtle()
if menu == "s":
    draw.forward(100)
    draw.left(90)
    draw.forward(100)
    draw.left(90)
    draw.forward(100)
    draw.left(90)
    draw.forward(100)
```



```
elif menu == "c":
    draw.circle(100)
elif menu == "r":
    draw.forward(100)
    draw.left(90)
    draw.forward(50)
    draw.left(90)
    draw.forward(100)
    draw.left(90)
    draw.forward(50)
elif menu == "t":
    draw.forward(100)
    draw.left(120)
    draw.forward(100)
    draw.left(120)
    draw.forward(100)
```

Task 2

- In your code add capability to fill the shape with some color. An example is given below



```
import turtle
t = turtle.Turtle()
t.fillcolor("blue")
t.begin_fill()
t.circle(50)
t.end_fill()
```

- Solution is ShapesSolutionColor.py

Launchpad Learning

That’s it for today

Thank You !!!!