Session Dos

Input

We can use the input() function to get input from the user.

```
print('Enter your name:')
x = input()                                # The input value is assigned to x
print('Hello, ' + x)                       # Concatenation of ‘Hello, ’ and x
```

Type casting

```
a = 2.5
b = int(a)
print(b)
```

Now let’s type cast the input:

```
x = int(input())                # The input value is taken as integer.
```

String Literals

String literals in python are surrounded by either single quotation marks, or double quotation marks.

`'hello'` is the same as `"hello"`.

Strings can be output to screen using the print function. For example: `print("hello")`.

Like many other popular programming languages, strings in Python are arrays of bytes representing unicode characters. However, Python does not have a character data type, a single character is simply a string with a length of 1. Square brackets can be used to access elements of the string.

1. Get the character at position 1 (remember that the first character has the position 0):

```
a = "Hello, World!"
print(a[1])
```

```
OUTPUT: e
```

2. Substring. Get the characters from position 2 to position 5 (not included):

```
b = "Hello, World!"
print(b[2:5])
```

```
OUTPUT: llo
```

3. The strip() method removes any whitespace from the beginning or the end:

```
a = " Hello, World! "
print(a.strip())              # returns "Hello, World!"
```

```
OUTPUT:  Hello, World!
```

4. The len() method returns the length of a string:

```
a = "Hello, World!"
print(len(a))
```

```
OUTPUT: 13
```

5. The lower() method returns the string in lower case:

```
a = "Hello, World!"
print(a.lower())
```

```
OUTPUT:hello, world!
```

6. The upper() method returns the string in upper case:

```
a = "Hello, World!"
print(a.upper())
```

```
OUTPUT: HELLO, WORLD!
```

7. The replace() method replaces a string with another string:

```
a = "Hello, World!"
print(a.replace("H", "J"))
```

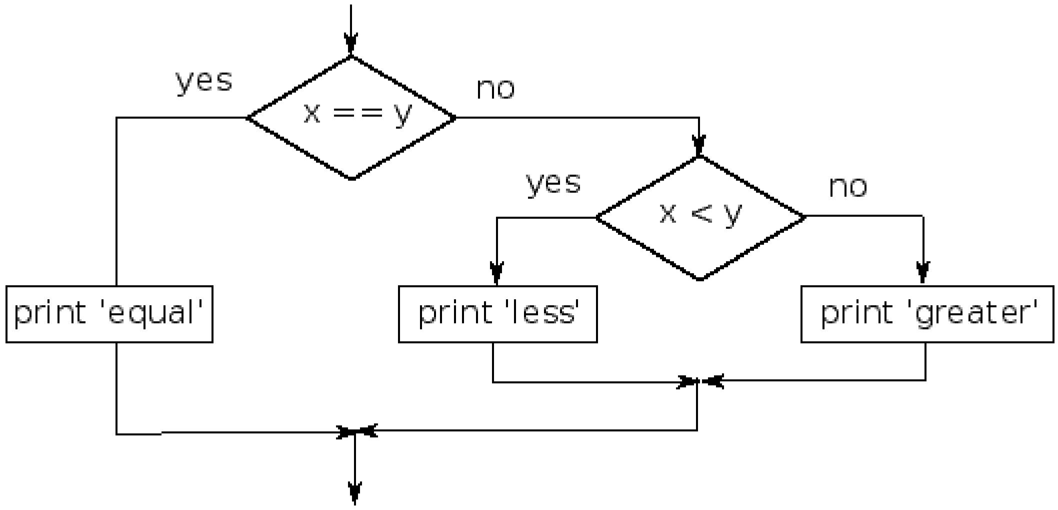
```
OUTPUT: Jello, World!
```

8. The split() method splits the string into substrings if it finds instances of the separator:

```
a = "Hello, World!"
print(a.split(","))           # returns ['Hello', ' World!']
```

```
OUTPUT:['Hello', ' World!']
```

Conditionals



Conditional operators:

Equals: a == b

Not Equals: a != b

Less than: a < b

Less than or equal to: a <= b

Greater than: a > b

Greater than or equal to: a >= b

```
a = 33
b = 200
if b > a:
    print("b is greater than a")

else

    print("a is greater than b")
```

But the output is wrong. Now fix it!

Problems

1. Find if the input number is odd or even.

2. Find if a number is negative or positive.

3. Find the maximum in three numbers.

Logical operators

and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

Problems

1. Print the age level of a given age. The range for the levels are:

0 – 15 = Child

16 – 35 = Youth

36 – Infinity = Senior

Input: 10

Output: Child

2. Write a Python program to input all sides of a triangle and check whether triangle is valid or not.

3. Determine the grades from an input number. The rage for the grades are -

Percentage	>=	90%	:	Grade	A
Percentage	>=	80%	:	Grade	B
Percentage	>=	70%	:	Grade	C
Percentage	>=	60%	:	Grade	D
Percentage	>=	40%	:	Grade	E
Percentage	<	40%	:	Grade	F