Lab 4

Python Operators

Lecturer: Hussien Omer AL _ Baiti

Python Arithmetic Operators

Arithmetic operators are used with numeric values to perform common mathematical operations:

Operator	Name	Example
+	Addition	
-	Subtraction	
*	Multiplication	
/	Division	
%	Modulus	
**	Exponentiation	
//	Floor division	

```
x = 5
print( x + 2)
print( x - 2)
print( x * 2)
print( x / 2)
print( x // 2)
print( x % 2)
print( x ** 2)
```

Python Assignment Operators

Assignment operators are used to assign values to variables

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3

Examples

```
x = 5
x += 2
x -= 2
x *= 2
x /= 2
x //= 2
x %= 2
x **= 2
print(x)
```

Python Comparison Operators

Comparison operators are used to compare two values:

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Python Logical Operators

Logical operators are used to combine conditional statements:

Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

number system ‘s Types

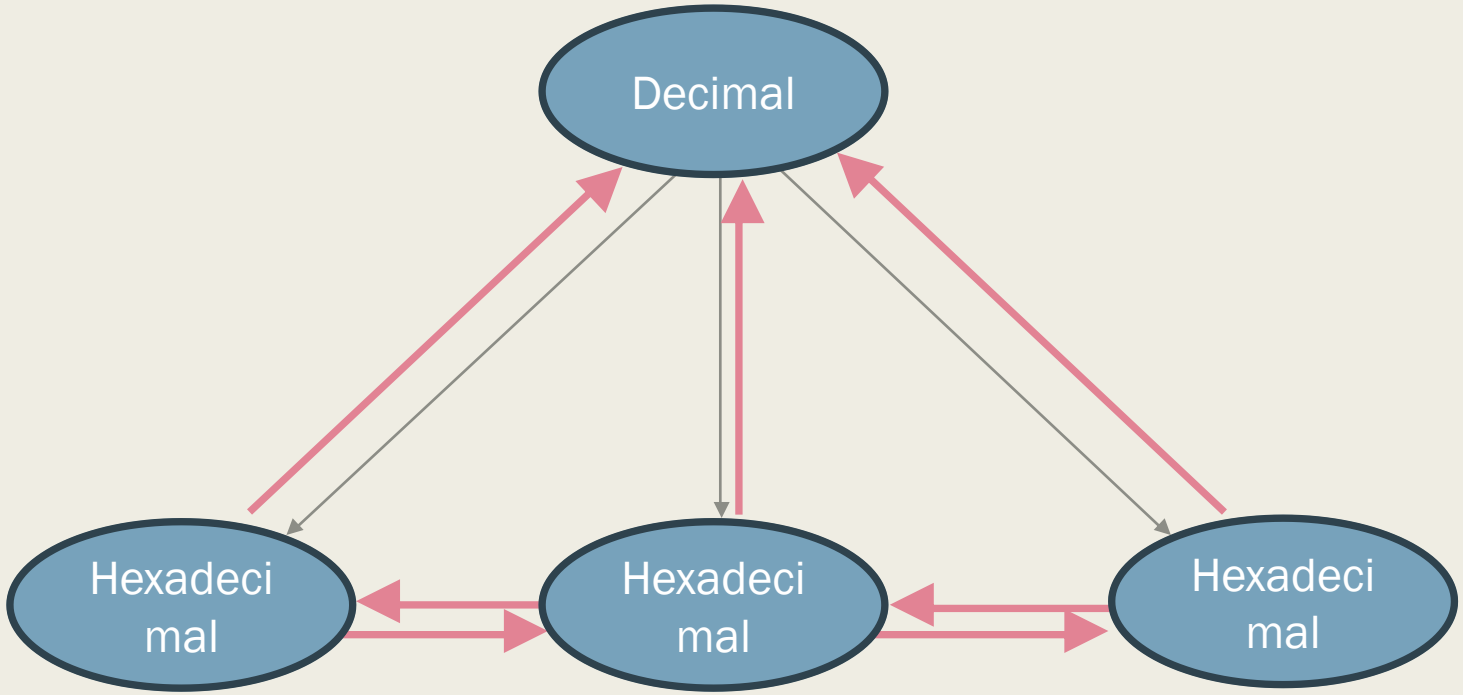
There are different types of number systems in math's like:

- **Decimal** number system → system base **10** → **0 - 9**
- **Binary** number system → system base **2** → **0 , 1**
- **Octal** number system → system base **8** → **0 - 7**
- **Hexadecimal** number system → system base **16** → **0 - 9** and **a - f**

Build in function of number systems

1. **Decimal** system : use **int ( )** function
2. **Binary** system : use **bin ( )** function
3. **Octal** system : use **oct ( )** function
4. **Hexadecimal** system : use **hex ( )** function

How can convert between numerical systems ?



Note : you can convert between decimal system into any type of systems directly, but if you need to convert between systems each other, should be convert into decimal then to another system by use special function.

Cont..

Output

```
x = 10
# convert into octal
x = oct(x)
print(x)
```

```
0o12
```

```
x = 10
# convert into hexadecimal
x = hex(x)
print(x)
```

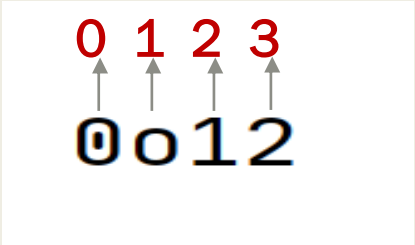
```
0xa
```

```
x = 10
# convert into binary
x = bin(x)
print(x)
```

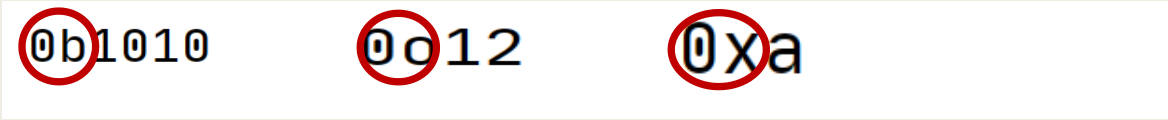
```
0b1010
```

Cont..

- Each element has address start from 0 into infinite positive value



- It’s a identity to confirm type of system number



- To show number without identity use Slicing concept , we will talk about it in details in next class (insha' Allah) , specific from second position

```
x = 10
# convert into binary
x = bin(x)[2:]
print(x)
```

Cont..

```
x = 10
x = bin(x)[2:]
print(x)
# to convert binary value into hexadecimal should be
# convert from binary to decimal then from decimal to hexadecimal
x = int(x,2)
x=hex(x)[2:]
print(x)
```

Select base system that need to convert it to decimal
2 is base of binary system that we need to convert it to decimal system .

Cont..

```
x = 10
x = hex(x)[2:]
print(x)
# to convert hexadecimal value into octal should be
# convert from hexadecimal to decimal then from decimal to octal
x = int(x,16)
x=oct(x)[2:]
print(x)
```

Select base system that need to convert it to decimal
16 is base of hexadecimal system that we need to convert it to decimal system .

Python Bitwise Operators

Bitwise operators are used to compare (binary) numbers:

Operator	Name	Description	Example
&	AND	Sets each bit to 1 if both bits are 1	x & y
|	OR	Sets each bit to 1 if one of two bits is 1	x | y
^	XOR	Sets each bit to 1 if only one of two bits is 1	x ^ y
~	NOT	Inverts all the bits	~x
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off	x << 2
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off	x >> 2

Steps of bitwise operator:

1. Convert items into binary system
2. Execute the operation ex: and , or , xor ..ect
3. Convert result into decimal system

```
x = 11
y = 12
print(x & y)
print(x | y)
print(x ^ y)
print( ~x )
print( x << 2 )
print( x >> 2 )
```

Example 1

```
x = 11
y = 12
print(x & y)
# convert items ( x , y ) into binary
x= bin(x)[2:] # x = 1011
y= bin(y)[2:] # y = 1100
print("x = ",x)
print("y = ",y)
# Execute the and operation on x,y will be equal 1000
# Convert result into decimal system
print(int("1000",2))
```

Example 2

```
x = 11
print(x << 2)
# convert items ( x ) into binary
x= bin(x)[2:] # x = 1011
print("x = ",x)
# Execute the left shift << operation on x will be equal 101100
# Convert result into decimal system
print(int("101100",2))
```

Example 3

```
x = 11
print(x >> 2)
# convert items ( x ) into binary
x= bin(x)[2:] # x = 1011
print("x = ",x)
# Execute the right shift >> operation on x will be equal 10
# Convert result into decimal system
print(int("10",2))
```

Home Work 1

- Write python Program and it’s algorithm to calculates root of number .

Notes : user read the number and root type


GOOD LUCK