

### # Develop Temp Conversion App

# Celsius to Fahrenheit ==>  $F = (C \times 9/5) + 32$ .

# Fahrenheit to Celsius ==>  $C = (F-32) \times (5/9)$

# Converting Celsius to Fahrenheit

```
celsius=float(input("Enter Temp in Celsius "))  
fahrenheit=(celsius*9/5)+32
```

```
print(f'{celsius:.2f}==> {fahrenheit:.2f}')
```

# Converting Fahrenheit to Celsius

```
fahrenheit=float(input("Enter Temp in fahrenheit"))  
celsius=(fahrenheit-32)*(5/9)
```

```
print(f'{fahrenheit:.2f} ==> {celsius:.2f}')
```

### Output

Enter Temp in Celsius 58

58.00==> 136.40

Enter Temp in fahrenheit136.40

136.40 ==> 58.00

# Write a Python program that calculates the total paycheck, factoring in regular hours and overtime.

```
hours_worked=int(input("Hours Worked "))
```

```
hourly_rate = 15
```

```
overtime_wage=22.5
```

```
paycheck=(40*hourly_rate)+((hours_worked-40)*overtime_wage)
```

```
print(paycheck)
```

### Output

Hours Worked 50

825.0

Hours Worked 40

600.0

## // Floor Division or Integer division

This operator is used for division. It is a binary operator and required 2 operands to perform operation

This operator divides two numbers and return result integer type

Division Operator required two operands

1. Dividend
2. Divisor

It will perform division and return Quotient (Integer Type)

This operator is applied only one numeric data types

### Example:

```
>>> a=5//2
>>> print(a,type(a))
2 <class 'int'>
>>> b=5/2
>>> print(b,type(b))
2.5 <class 'float'>
>>> c=4//2
>>> print(c,type(c))
2 <class 'int'>
```

Return the **floor** of x, the largest integer less than or equal to x

```
>>> c=-5//2
>>> print(c)
-3
```

### Example:

# Write a program to delete/remove last digit  
# of input integer value

```
num=int(input("Enter any number "))
num1=num//10
```

```
print(f'Original Number {num}')
print(f'After Deleting Digit {num1}')
```

**Output**

Enter any number 124  
Original Number 124  
After Deleting Digit 12

Enter any number -124  
Original Number -124  
After Deleting Digit -13

**Example:**

```
>>> a=5.0//2
>>> print(a)
2.0
>>> b=5//2.0
>>> print(b)
2.0
```

**% (Modulo Operator)**

This operator is used for division. This operator after division it returns remainder. It is a binary operator and required 2 operands to perform operation.

**Example:**

```
>>> n1=5%3
>>> print(n1)
2
>>> n2=10%6
>>> print(n2)
4
>>> n3=4%2
>>> print(n3)
0
>>> n4=5%2
>>> print(n4)
1
```

**\*\* Exponent Operator (OR) Power of Operator**

It is binary operator and required 2 operands to perform operation

This operator is used for finding power of any number

```
>>> res1=5**2
>>> print(res1)
25
>>> res2=3**3
>>> print(res2)
27
>>> res3=2**8
>>> print(res3)
256
```

## Operator Precedence

Operator precedence tells order of execution of operators. The following table summarizes the **operator precedence** in Python, from highest **precedence** (most binding) to lowest **precedence** (least binding). **Operators** in the same box have the same precedence. Unless the syntax is explicitly given, **operators** are binary. **Operators** in the same box group left to right (except for exponentiation and conditional expressions, which group from right to left).

Operator	Description
(expressions...), [expressions...], {key: value...}, {expressions...}	Binding or parenthesized expression, list display, dictionary display, set display
x[index], x[index:index], x(arguments...), x.attribute	Subscription, slicing, call, attribute reference
<a href="#">await x</a>	Await expression
**	Exponentiation
+x, -x, ~x	Positive, negative, bitwise NOT
*, @, /, //, %	Multiplication, matrix multiplication, division, floor division, remainder
+, -	Addition and subtraction

<<, >>	Shifts
&	Bitwise AND
^	Bitwise XOR
	Bitwise OR
<a href="#">in</a> , <a href="#">not in</a> , <a href="#">is</a> , <a href="#">is not</a> , <, <=, >, >=, !=, ==	Comparisons, including membership tests and identity tests
<a href="#">not x</a>	Boolean NOT
<a href="#">and</a>	Boolean AND
<a href="#">or</a>	Boolean OR
<a href="#">if</a> – else	Conditional expression
<a href="#">lambda</a>	Lambda expression
:=	Assignment expression

#### Precedence of Arithmetic Operator

**
*, @, /, //, %
+, -