

# All about for Loops

☰ Contents

Print to PDF ►

[Task 1](#)

[Task 2](#)

[Task 3](#)

[Task 4](#)

Let us understand how we can loop through using `for` in Python.

- Go through range between 2 integers (5 and 10) and print them. We can use range function to get range between two integers.

```
range(5, 11)
```

```
range(5, 11)
```

```
list(range(5, 11))
```

```
[5, 6, 7, 8, 9, 10]
```

```
list(range(5, 11, 2))
```

```
[5, 7, 9]
```

```
for i in range(5, 11): print(i)
```

```
5
6
7
8
9
10
```

- Create a list of 7 elements and print alternate numbers starting from 1. `[1, 6, 8, 3, 7, 2, 9]`
- In this example we are using list. It will be covered extensively at a later point in time. `list` is one of the core Python Data Structures.

```
l = [1, 6, 8, 3, 7, 2, 9]
```

```
type(l)
```

```
list
```

```
cnt = 0
for i in l:
    if cnt % 2 == 0: # checking if cnt is even or not
        print(i)
    cnt += 1 # incrementing cnt by 1
```

```
1
8
7
9
```

- Go through range between 2 integers (5 and 15) and print even numbers.

```
for i in range(5, 16):
    if i % 2 == 0:
        print(i)
```

```
6
8
10
12
14
```

- Iterate through a list of months and print them.

```
months = ['January', 'February', 'March']
for month in months:
    print(month)
```

```
January
February
March
```

```
import calendar
list(calendar.month_name)
```

```
['',
 'January',
 'February',
 'March',
 'April',
 'May',
 'June',
 'July',
 'August',
 'September',
 'October',
 'November',
 'December']
```

```
for month in list(calendar.month_name)[1:]:
    print(month)
```

```
January
February
March
April
May
June
July
August
September
October
November
December
```

# Task 1

Take the list of ages and determine the category of the baby.

- Print **New Born or Infant** till 6 months.
- Print **Toddler** from 7 months to 18 months.
- Print **Grown up** from 19 months to 144 months.
- Print **Youth** from 145 months to 216 months

```
ages = [7, 3, 15, 145, 217, 18]
```

```
for age in ages:
    if age <= 6:
        print(f'Kid with {age} months age is "New Born or Infant"')
    elif age > 6 and age <= 18:
        print(f'Kid with {age} months age is "Toddler"')
    elif age > 18 and age <= 144:
        print(f'Kid with {age} months age is "Grown up"')
    elif age > 144 and age <= 216:
        print(f'Kid with {age} months age is "Youth"')
    else:
        print(f'Kid with {age} months age is "Adult"')
```

```
Kid with 7 months age is "Toddler"
Kid with 3 months age is "New Born or Infant"
Kid with 15 months age is "Toddler"
Kid with 145 months age is "Youth"
Kid with 217 months age is "Adult"
Kid with 18 months age is "Toddler"
```

## Task 2

Check if each number in list of integers is even or divisible by 3.

```
ns = [1, 6, 2, 7, 9, 11, 3, 4]
```

```
for n in ns:
    if n % 2 == 0 or n % 3 == 0:
        print(f'Number {n} is even or divisible by 3')
```

```
Number 6 is even or divisible by 3
Number 2 is even or divisible by 3
Number 9 is even or divisible by 3
Number 3 is even or divisible by 3
Number 4 is even or divisible by 3
```

## Task 3

Check if the number in list of integers is even and divisible by 3

```
ns = [1, 6, 2, 7, 9, 11, 3, 4]
```

```
for n in ns:
    if n % 2 == 0 and n % 3 == 0:
        print(f'Number {n} is even or divisible by 3')
    elif n % 3 == 0:
        print(f'Number {n} is divisible by 3 but not even')
    elif n % 2 == 0:
        print(f'Number {n} is even but not divisible by 3')
```

```
Number 6 is even or divisible by 3
Number 2 is even but not divisible by 3
Number 9 is divisible by 3 but not even
Number 3 is divisible by 3 but not even
Number 4 is even but not divisible by 3
```

## Task 4

Check if each salary in list of salaries is valid. If the salary is negative or 0, then print **Salary {salary\_amount} is invalid**. If the salary is **None** then print **Salary is Not Available**.

```
sals = [100.0, 1000.0, 0.0, -100, 1200.0, None, -500]
```

```
for sal in sals:
    if sal == None:
        print('Salary is Not Available')
    elif sal <= 0:
        print(f'Salary {sal} is invalid')
```

```
Salary 0.0 is invalid
Salary -100 is invalid
Salary is Not Available
Salary -500 is invalid
```

```
for sal in sals:
    if not sal:
        print('Salary is Not Available')
    elif sal <= 0:
        print(f'Salary {sal} is invalid')
```

```
Salary is Not Available
Salary -100 is invalid
Salary is Not Available
Salary -500 is invalid
```

## Note

We can also use list of tuples with employee id and salary. You will see examples like this as part of manipulating collections later.

```
emp_sals = [  
    (1, 100.0),  
    (2, 1000.0),  
    (3, 0),  
    (4, -100),  
    (5, 1200.0),  
    (6, None),  
    (7, -500)  
]
```

```
for emp_sal in emp_sals:  
    if emp_sal[1] == None:  
        print(f'Salary for employee with id {emp_sal[0]} is Not Available')  
    elif emp_sal[1] <= 0:  
        print(f'Salary {emp_sal[1]} for employee with id {emp_sal[0]} is invalid')
```

```
Salary 0 for employee with id 3 is invalid  
Salary -100 for employee with id 4 is invalid  
Salary for employee with id 6 is Not Available  
Salary -500 for employee with id 7 is invalid
```