

Python

```
# you can write any coment here as long as it is a single line.  
# no need to declare variables.  
name= 'Hanna'  
age = 14
```

```
height= 1.75
```

```
print(type(name))  
print(type(age))  
print(type(height))
```

```
x = 1  
x = 'char'  
print()
```

tuple

- ▼ list.
- ▼ it cant be changed.
- ▼

```
#tuple  
names=['hanna', 'kebede', 'yakob']  
print(type(names))
```

```
#to add a string to the array
names.append("Hailu")

#to change a certain string in an array
names.insert(3, 'alex')
```

Set

```
#set
#you use cursive brace to list out items.
name1={'Hanna', 'yosef', 'kebede'}
print(type(name1))
```

Casting

casting means changing the data type of a certain object.

```
height= 1.74
#initially height data type was float but if we want to change it to int
height= int(1.74)
```

Dictionary

```
name1_info={'name': 'Yosef' , 'age': 24 , 'section' : 'B' , 'Hobbies': ['reading', 'gym'],
            'skills' : ['java' , 'HTML' , 'Javascript' , 'python']}
}
```

```
#to access a specific array
print(name1_info['height'])
```

Flow control

```
for i in range(0 , 12):
    print(i)
#to access the index of an array
for index, value in enumerate(name1_info):
    print(index)

#to access the value of index in an array
for index, value in enumerate(name1_info):
    print(value)
```

If else

```
age = 45
if age > 18 :
    print('adult')

elif age < 18:
    print('underage')

else:
    print('Error')
```

Function

```
#Function
#needs to be called in order to get executed
```

```

def greet():
    print('Hello World')

def greet_with_name(name):
    print(f'Hello {name}')

greet()

greet_with_name('hanna')

#to add
def greet_with_name(name, age, Last_name = 'Kebede'):
    print(f'Hello {name} age : {age} last name : {Last_name}')

print('hanna' , age = 56)

#positional arguments
# keyword arguments
# default arguments
# string formatting using f string

def greet_name(*numbers):
    print(numbers)

greet_name('hanna' , 'kebede', 'chala' )

```

Map

```

#a function that squares numbers
numbers = [3,4,34,4,34,3,5,657,6,7]

```

```

square_value = []

def square_List(list_element):
    for number in list_element:
        square_value.append(number * number)
    print(square_value)

square_List(numbers)

#lambda par:exp squaring function.

square = lambda number: number * number

print(square(5))

#fun that checks if a number is even/odd.
check = lambda number: 'even number' if number % 2 == 0 else 'odd number'

print(check(69))

#map(function, iterables/variables)
print(list(map(lambda number: number * number, numbers)))

#filter the odd value from a given iterables
#filter(function, iterables)
new_value = filter(lambda number: True if number % 2==0 else False, numbers)
print(list(new_value))

```

Object Oriented

```

class student :
    name = ''
    student_id = ''

```

```
stu1 = student()

stu1.student_id = 4587
stu1.name = 'Hanna'

stu2 = student()

stu2.student_id = 8974
stu2.name = 'Aaron'

stu3 = student()

stu3.student_id = 5478
stu3.name = 'Jaquelin'
```

Example

```
class flight:
    destination = ''
    source = ''
    estimated_time = ''
    passenger_list = []
    capacity = 5

    def __init__(self, source, destination, capacity) -> None:
        self.source = source
        self.capacity = capacity
        self.destination = destination

    def add_passenger(self, full_name:str):
        if len(self.passenger_list) >= self.capacity:
```

```

        print(f'sorry {full_name}the flight is fully booked
else:
    self.passenger_list.append(full_name)
    print(f'{full_name} has been added to the passenger:

def view_passenger(self):
    print(self.passenger_list)

def remove_passenger(self, full_name):
    if full_name in self.passenger_list:
        index = self.passenger_list.index(full_name)
        print(f'{full_name} has been removed from the passer

flight1 = flight('ADB','DXB','8')

names = ['Hanna' , 'Abel' , 'Alex' , 'Bereket' , 'Mati']

flight1.add_passenger(names)

flight1.view_passenger

```

Inheritance

```

class product:
    price = ''
    name = ''
    brand = ''

class Electronics(product):
    model_number = ''

```

```
class cloth(product):  
    size = ''  
  
elec1 = Electronics()  
elec1.model_number = 'Samsung'  
  
#to add or change price  
  
def set_price(self , amount)  
    self.price = amount  
    print(amount)
```