```
>>> n = 24
>>> if (n % 2) == 0:
        print("even")  # even
else:
        print("odd")


even
>>> n = 35
>>> if (n % 2) == 0:
        print("even")  # even
else:
        print("odd")


odd
>>> 2**3
8
>>> #2**n  n is an integer
>>> import math
>>> help(math.log)
Help on built-in function log in module math:

log(...)
    log(x, [base=math.e])
    Return the logarithm of x to the given base.

    If the base not specified, returns the natural logarithm (base e) of x.

>>> 2 ** 3
8
>>> 2 * 2 *2
8
>>> 4 *4
16
>>> 4 ** 4
256
>>> 4 * 4 * 4 * 4
256
>>> math.log(8,2)
3.0
>>> math.log(10,2)
3.3219280948873626
>>> # even or not and then whether it is a power of 2
>>> n = 64
>>> if (n%2) == 0:
        if(log(n,2).is_integer()):
                print("n is even and a power of 2")
        else:
                print("n is even but not a power of 2")
else:
        # odd
        print("n is odd")


Traceback (most recent call last):
  File "<pyshell#30>", line 2, in <module>
    if(log(n,2).is_integer()):
NameError: name 'log' is not defined
>>> if (n%2) == 0:
        if(math.log(n,2).is_integer()):
                print("n is even and a power of 2")
        else:
                print("n is even but not a power of 2")
else:
        # odd
        print("n is odd")


n is even and a power of 2
>>> if (n%2) == 0:
        if(math.log(n,2).is_integer()):
                print("n is even and a power of 2")
                print(f"{n} is the {math.log(n,2)}th power of 2")
        else:
                print("n is even but not a power of 2")
else:
        # odd
        print("n is odd")


n is even and a power of 2
64 is the 6.0th power of 2
>>> if (n%2) == 0:
        if(math.log(n,2).is_integer()):
                print("n is even and a power of 2")
                print(f"{n} is the {int(math.log(n,2))}th power of 2")
        else:
                print("n is even but not a power of 2")
else:
        # odd
        print("n is odd")


n is even and a power of 2
64 is the 6th power of 2
>>> 2 ** 6
64
>>> pow(2, 6)
64
>>> math.log(64,2)
6.0
>>> n = 44
>>> if (n%2) == 0:
        if(math.log(n,2).is_integer()):
                print("n is even and a power of 2")
                print(f"{n} is the {int(math.log(n,2))}th power of 2")
        else:
                print("n is even but not a power of 2")
else:
        # odd
        print("n is odd")


n is even but not a power of 2
>>> math.log(n,2)
5.459431618637297
>>> n = 3
>>> if (n%2) == 0:
        if(math.log(n,2).is_integer()):
                print("n is even and a power of 2")
                print(f"{n} is the {int(math.log(n,2))}th power of 2")
        else:
                print("n is even but not a power of 2")
else:
        # odd
        print("n is odd")


n is odd
>>> if (n%2) == 0:
        m = 0
```

```
    if(math.log(n,2).is_integer()):
        m = "hello"
        print("n is even and a power of 2")
        print(f"{n} is the {int(math.log(n,2))}th power of 2")
    else:
        print("n is even but not a power of 2")
else:
    m= 3.4
    # odd
    print("n is odd")


n is odd
>>> m
3.4
>>> n = 2
>>> m
3.4
>>> if (n%2) == 0:
    m = 0
    if(math.log(n,2).is_integer()):
        m = "hello"
        print("n is even and a power of 2")
        print(f"{n} is the {int(math.log(n,2))}th power of 2")
    else:
        print("n is even but not a power of 2")
else:
    m= 3.4
    # odd
    print("n is odd")


n is even and a power of 2
2 is the 1th power of 2
>>> m
'hello'
>>> n =6
>>> if (n%2) == 0:
    m = 0
    if(math.log(n,2).is_integer()):
        m = "hello"
        print("n is even and a power of 2")
        print(f"{n} is the {int(math.log(n,2))}th power of 2")
    else:
```

```
                print("n is even but not a power of 2")
else:
        m= 3.4
        # odd
        print("n is odd")


n is even but not a power of 2
>>> if (n%2) == 0:
        m = 0
        if(math.log(n,2).is_integer()):
                m = "hello"
                print("n is even and a power of 2")
                print(f"{n} is the {int(math.log(n,2))}th power of 2")
        else:
                m = [1,"a"]
                print("n is even but not a power of 2")
else:
        m= 3.4
        # odd
        print("n is odd")


n is even but not a power of 2
>>> n = 6
>>> if (n%2) == 0:
        m = 0
        if(math.log(n,2).is_integer()):
                m = "hello"
                print("n is even and a power of 2")
                print(f"{n} is the {int(math.log(n,2))}th power of 2")
        else:
                m = [1,"a"]
                print("n is even but not a power of 2")
else:
        m= 3.4
        # odd
        print("n is odd")


n is even but not a power of 2
>>> m
[1, 'a']
>>> q = (1, 2, 3)
>>> w = [1,2,3]
>>> w[0] = 100
>>> w
[100, 2, 3]
>>> q[0] = 100
Traceback (most recent call last):
  File "<pyshell#68>", line 1, in <module>
    q[0] = 100
TypeError: 'tuple' object does not support item assignment
>>> q[0]
1
>>> q[0:2]
(1, 2)
>>> w
[100, 2, 3]
>>> tuple
<class 'tuple'>
>>> x = tuple(w)
>>> x
(100, 2, 3)
>>> t = ()
>>> t
()
>>> type(t)
<class 'tuple'>
>>> s = (1)
>>> type(s)
<class 'int'>
>>> s = (1,)
>>> type(s)
<class 'tuple'>
>>> s = (1,)  # remember to include a , after the value when initializing a
>>>
>>>

>>>


>>>
>>>
>>> # with single element
>>> #########################################
>>> "" # delimter
''
```

```
>>> #  str is used to represent text
>>> ' ' #
' '
>>> " "
' '
>>> len("asasda")
6
>>> "'"
"'"
>>> "\""
'"'
>>> print("\n")



>>> print("India's score today is 349")
India's score today is 349
>>> print('India's score today is 349')

SyntaxError: invalid syntax
>>> print('India\'s score today is 349')
India's score today is 349
>>> print(""" Hey this is line 1 """)
 Hey this is line 1
>>> 
======= RESTART: C:/Users/Python/Python39/print_multi_demo.py =======
 Hey this is line 1
 line 2
 line 3
>>> s = "Apple"
>>> s.lower()
'apple'
>>> s
'Apple'
>>> s.upper()
'APPLE'
>>> s
'Apple'
>>> s = s.lower()
>>> s
'apple'
>>> s = "Apple"
>>> s = s.upper()
>>> s
'APPLE'
```

```
>>> s = "Apple"
>>> c = "is the name of a company"
>>> f = "is the name of a fruit"
>>> s + c
'Appleis the name of a company'
>>> s + " " + c
'Apple is the name of a company'
>>> s + " " + f
'Apple is the name of a fruit'
>>> d = "Today is wednesday    "
>>> d
'Today is wednesday    '
>>> d = "     Today is wednesday    "
>>> d.rstrip()  # removes Trailing white space
'     Today is wednesday'
>>> d
'     Today is wednesday    '
>>> d.lstrip() # Removes Leading white space
'Today is wednesday    '
>>> d.strip()  # Removes both leading trailing white space
'Today is wednesday'
>>> dir(d)  # lists all the available methods for a datatype
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__',
'__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii',
'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper',
'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffix', 'replace', 'rfind',
'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title',
'translate', 'upper', 'zfill']
>>> d
'     Today is wednesday    '
>>> g = "good"
>>> h = "hood"
>>> g.endswith('d')
True
>>> g.endswith('o')
False
>>> g.endswith('od')
True
>>> g.startswith('g')
True
```

```
>>> g.startswith('h')
False
>>> h.startswith('h')
True
>>> "h" in h
True
>>> "o" in h
True
>>> "oo" in h
True
>>> g = "goo"
>>> g.endswith('o')
True
>>> g.endswith('oo')
True
>>> g.startswith('g')
True
>>> g.startswith('go')
True
>>> "oo" in "good"
True
>>> "og" in "good"
False
>>> "G" in "Good"
True
>>> "good".find('d')
3
>>> "good".find('o')
1
>>> "good"[1]
'o'
>>> "good".index('o')
1
>>> "good".index('1')
Traceback (most recent call last):
  File "<pyshell#151>", line 1, in <module>
    "good".index('1')
ValueError: substring not found
>>> "good".find('1')
-1
>>> "good".rfind("o")
2
>>> "good".rindex("o")
2
```

```
>>> "good".rindex("9")
Traceback (most recent call last):
  File "<pyshell#155>", line 1, in <module>
    "good".rindex("9")
ValueError: substring not found
>>> "good".rfind("9")
-1
>>> len("dad")
3
>>> "dad".upper()
'DAD'
>>> "adasdsad"[-1]
'd'
>>> len("adasdsad")
8
>>> l = len("adasdsad")
>>> s = "adasdsad"
>>> l = len(s)
>>> l
8
>>> s[l - 1]
'd'
>>> s[-1]
'd'
>>> l - 1
7
>>> s[0]
'a'
>>> s[-l]
'a'
>>> s
'adasdsad'
>>> s[2:5]
'asd'
>>> s[:]
'adasdsad'
>>> s[2:]
'asdsad'
>>> t= "abcdefgh"
>>> t[:]
'abcdefgh'
>>> t[1]
'b'
>>> t[1:]
```

```
'bcdefgh'
>>> t[3]
'd'
>>> t[:3]
'abc'
>>> t[1:4]
'bcd'
>>> t[1:]
'bcdefgh'
>>> t[1:6]
'bcdef'
>>> t[1:6:2]
'bdf'
>>> t[::2] # values at even indices
'aceg'
>>> t[1::2] # values at even indices
'bdfh'
>>> t[::3] # values at even indices
'adg'
>>> f'{t}'
'abcdefgh'
>>> '{t}'
'{t}'
>>> 't'
't'
>>> f't'
't'
>>> f'{1+5}'
'6'
>>> str(3)
'3'
>>> str(3.90)
'3.9'
>>> str([121,213])
'[121, 213]'
>>>
```