Arithmetic Operation and String Manipulation

Arithmetic operation

Operator:	Operation:
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder
//	Floor division
**	Exponent

Arithmetic operation example

```
x = 8
y = 3

print("Addition:\t", x, "+", y, "=", x+y)
print("Subtraction:\t", x, "-", y, "=", x-y)
print("Multiplication:\t", x, "*", y, "=", x*y)
print("Division:\t", x, "/", y, "=", x/y)
print("Floor division:\t", x, "//", y, "=", x//y)
print("Reminder:\t", x, "%", y, "=", x%y)
print("Exponent:\t", x, "**", y, "=", x**y)
```

Program output

```
Addition:         8 + 3 = 11
Subtraction:      8 - 3 = 5
Multiplication:   8 * 3 = 24
Division:         8 / 3 = 2.6666666666666665
Floor division:   8 // 3 = 2
Reminder:         8 % 3 = 2
Exponent:         8 ** 3 = 512
```

Assigning values

Operator:	Example:	Equivalent:
=	a = b	a = b
+=	a += b	a = (a + b)
-=	a -= b	a = (a - b)
*=	a *= b	a = (a * b)
/=	a /= b	a = (a / b)
%=	a %= b	a = (a % b)
//=	a //= b	a = (a // b)
**=	a **= b	a = (a ** b)

Assigning value example

```
b=3
a=b
print("Assign value:\t\t", "a=", a)
a+=b
print("Add & assign:\t\t", "a=", a, " \t\t(3+3)" )
a-=b
print("Subtract & assign:\t", "a=", a, "\t\t(6-3)" )
a*=b
print("Multiply & assign:\t", "a=", a, "\t\t(3*3)" )
a/=b
print("Divide & assign:\t", "a=", a, "\t(9/3)" )
a%=b
print("Remainder & assign:\t", "a=", a, "\t(3%3)" )
b**=3
print("Exponent & assign:\t", "b=", b, "\t\t(3**3)")
```

Assigning value example

Program output

```
Assign value:              a= 3
Add & assign:              a= 6          (3+3)
Subtract & assign:         a= 3          (6-3)
Multiply & assign:         a= 9          (3*3)
Divide & assign:           a= 3.0        (9/3)
Remainder & assign:        a= 0.0        (3%3)
Exponent & assign:         b= 27         (3**3)
```

String Manipulation

Operator:	Description:	Example:
+	Concatenate – join strings together	‘Hello’ + ‘Mike’
*	Repeat – multiply the string	‘Hello’ * 2
[ ]	Slice – select a character at a specified index position	‘Hello’ [0]
[:]	Range Slice – select characters in a specified index range	‘Hello’ [0 : 4]
in	Membership Inclusive – return True if character exists in the string	‘H’ in ‘Hello’
not in	Membership Exclusive – return True if character doesn’t exist in string	‘h’ not in ‘Hello’

String manipulation example

```
s1 = "Hello"
s2 = "World"
print("s1+s2 is:\t", s1+s2)
print("s1*2 is:\t", s1*2)
print("s1[1] is:\t", s1[1])
print("s1[1:3] is:\t", s1[1:3])
print("Is H in s1?\t", ("H" in s1))
print("Is h in s1?\t", "h" in s1)
print("is h not in s1?\t", "h" not in s1 )
```

Program output

```
s1+s2 is:          HelloWorld
s1*2 is:           HelloHello
s1[1] is:          e
s1[1:3] is:        el
Is H in s1?        True
Is h in s1?        False
is h not in s1?    True
```

Addition string manipulation methods

Method:	Description:
replace(*old, new*)	Replace all occurrences of *old* with *new*
count(*sub*)	Return the number of occurrences of *sub*
find(*sub*)	Return the index number of the first occurrence of *sub* or return -1 if not found
isdigit( ) **isdecimal( )**	Return True if string contains only digits or decimals – otherwise return False
isspace( )	Return True if string contains only whitespace – otherwise return False
lstrip( ) **rstrip ( )** **strip( )**	Remove leading whitespace, trailing whitespace, or both leading and trailing whitespace respectively
center(*w, c*)	Pad string each side to total column width *w* with character *c* (default is space)
ljust(*w, c*) **rjust(*w, c*)**	Pad string to right or left respectively to total column width *w* with character *c*

String method example

```
s1 ="Learning Python program in easy steps"
s2=s1.replace("program", "coding")
print(s2)
gnum = s1.count("g")
print("The number of occurance of g is", gnum)
p_index = s1.find("P")
print("The index of the first occurance of P is", p_index)
```

Program output

```
Learning Python coding in easy steps
The number of occurance of g is 2
The index of the first occurance of P is 9
```

Additional string method example

```
a="12"
b="12.3"
c="number"
d="    "
print("Is a digit?\t", a.isdigit() )
print("Is b decimal?\t", a.isdecimal() )
print("Is b digit?\t", b.isdigit() )
print("Is b decimal?\t", b.isdecimal() )
print("Is c digit?\t", c.isdigit() )
print("Is d space?\t", d.isspace() )
```

Program output

```
Is a digit?     True
Is b decimal?   True
Is b digit?     False
Is b decimal?   False
Is c digit?     False
Is d space?     True
```

Additional string method example

```
s1="      Hello World      "
s2="|"
s3=s1.lstrip()
print(s2+s3+s2)
s3=s1.rstrip()
print(s2+s3+s2)
s3=s1.strip()
print(s2+s3+s2)
```

Program output

```
|Hello World      |
|      Hello World|
|Hello World|
```

Additional string method example

```
s1="Hello Word"
s2=s1.center(20, "*")
print(s2)
s2=s1.ljust(20, "*")
print(s2)
s2=s1.rjust(20, "*")
print(s2)
```

Program output

```
*****Hello Word*****
Hello Word**********
**********Hello Word
```

Today's Challenge 1

- In the guessing game we previously discussed, if the user enter decimals or texts, the program will run into problems.
- Can you improve the program such that it can remind user to enter integers when the user enters decimals or texts?

```
Enter your number (1-15): q
Please enter an integer number

Enter your number (1-15): 12.3
Please enter an integer number

Enter your number (1-15): 8
Your guess is too large
```

Today's Challenge 2

- In the challenge problem of the last section, the following codes are used to construct the tree top. Could you rewrite this portion of the program using string center method?

```
# print the tree top
for x in range(1, treetop+1):
    line=""
    for k in range(1, treetop-x+1):
        line=line+" "
    for k in range(1, 2*x):
        line=line+"A"
    print(line)
```

```
          A
         AAA
        AAAAA
       AAAAAAA
      AAAAAAAAA
     AAAAAAAAAAA
    AAAAAAAAAAAAA
   AAAAAAAAAAAAAAA
  AAAAAAAAAAAAAAAAA
 AAAAAAAAAAAAAAAAAAA
          H
          H
          H
          H
          H
--------------------
```

Solution for Challenge Problem 1

```
import random
num = random.randint(1, 15)
cnt=1
correct = False

while((not correct) and cnt<=4 ):
    guess = input("Enter your number (1-15): ")

    if guess.isdigit():
        guess_num = int(guess)
        if (guess_num>num):
            print("Your guess is too large")
        elif (guess_num<num):
            print("Your guess is too small")
        else:
            print("Your guess is correct!")
            correct = True
        cnt = cnt+1
    else:
        print("Please enter an integer number")

if (correct):
    print("You win the game")
else:
    print("You lose the game. The number is", num)
```

Solution for Challenge Problem 2

```
tree_top_height = input("Please enter the height of tree top:")
tree_trunk_height =  input("please enter the height of the tree trunk:")
treetop = int(tree_top_height)
treetrunk = int(tree_trunk_height)

# print the tree top
for x in range(1, treetop+1):
    Astr="A"*(2*x-1)
    line=Astr.center(2*treetop+1)
    print(line)

# print the tree chunk
Hstr="H"
line = Hstr.rjust(treetop+1)

for x in range(treetrunk):
    print(line)

# print the tree base
line = ""
for x in range(2*treetop+1):
    line = line+"-"
print(line)
```