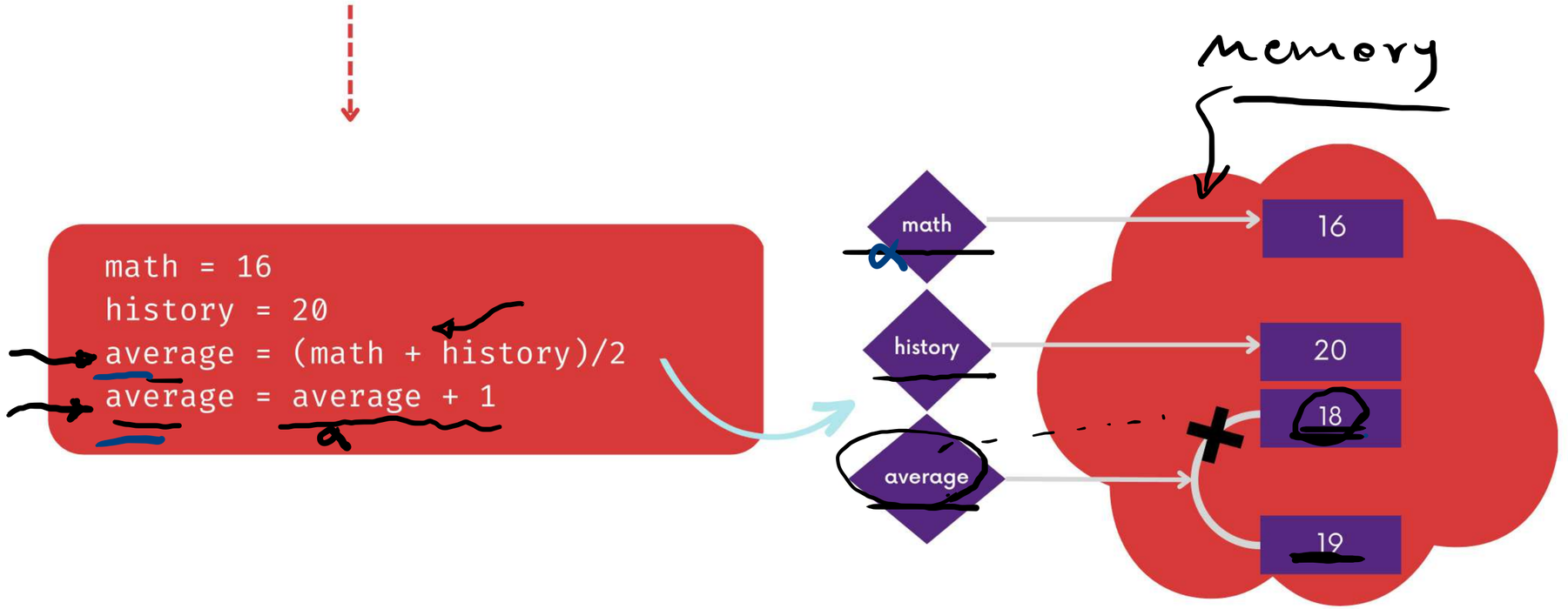
Changing Binding

- Can re-bind variable names using new assignment statements.
- Previous value may still stored in memory but lost the handle for it
- Value for area does not change until you tell the computer to do the calculation again.



Strings

- Sequence of characters, Letters, Special Characters, Spaces, Digits
- enclose in quotation marks or single quotes

```
hi = "hello there"
```

- Concatenate Strings

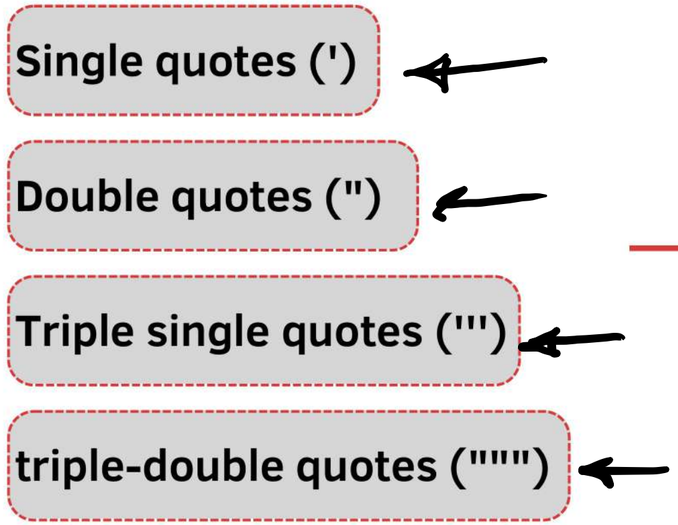

```
hi = "hello there"
name = "ana"

greet = hi + name
#out: hello thereana

greeting = hi + " " + name
#out: hello there ana

operat = hi + " " + name * 3
#out: hello there anaanaana
```

String Literals




```
s = 'This is a string'
print(s)

s = "Another string using double quotes"
print(s)

s = ''' string can span multiple line '''
print(s)

s = """ string can span multiple line """
print(s)
```

Input

- Prints whatever is in the quotes
- User types in something and hits enter
- Binds that value to a variable

```
text = input("Type anything ...")
print(text)
#out: input text
```

- input gives you a string so must cast if working with numbers

```
num = int(input("Type a number ..."))
print(num)
#out: input num
```

age = input('age')

Typecasts:

String

age + 5 => error

str int

Exercises 02

Operators
900

Operators

Exercise 01

- Get a number as circle diameter, and calculate the circle area!

Points:

- Use **input method** to get circle diameter!
- Use **str.format()** method to print the output!
- Output must be same as following example, exactly!

```
Example:  input: 18 , output (print): Circle area is 254.469
```

Operators

Exercise 02

- Get width, height and length ,then calculate and print the surface area and volume of a cuboid!

Points:

- Use **input method** to get the inputs!
- Use **str.format()** method to print the output!
- Output must be same as following example, exactly!

```
Example:  input1: 18, input2:4, input3:5 , output (print): Volume of cuboid is 360.00 and Surface area
          of cuboid is 364.000!
```

Operators

Exercise 03

- input(2 number) and print sum, division, subtraction, multiplication

Points:

- Use **input method** to get circle diameter!
- Use **str.format()** method to print the output!
- Output must be same as following example, exactly!

```
Example:  input_1: 8, input_2: 4 , output: sum = 12, division=2, subtraction=4, multiplication=32
```

Operators

Exercise 04

- Get two string, concatenate them and print the result!

Points:

- insert a space between two strings.
- Use **input method** to get circle diameter!
- Use **str.format()** method to print the output!
- Output must be same as following example, exactly!

```
Example:  input_1: string_1 , input_2: string_2 , output: result is "string_1 string_2"
```

Operators

Exercise 05

- Write a Python program to convert Fahrenheit to Celcius.

Points:

- Use **input method** to get Fohrenheit!
- Use **str.format()** method to print the output!
- Output must be same as following example, exactly!

```
Example: input: 86 , output: 86 degree Fahrenheit is equal to 30.0 degree Celsius.
```

Operators

Exercise 06 (Search!)

- Write a code to create following pattern!

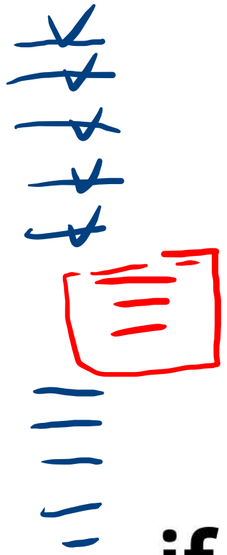
```
#
##
###
####
#####
######
#######
```

print('#')

print('##')

loop

Control flow &Iteration



if Statement

- <condition> has a value True or False
- Evaluate expressions in that block if <condition> is True

```
if <condition>:
    <expression>
    <expression>
    ...
```

```
if <condition>:
    <expression>
    <expression>
    ...
else:
    <expression>
    <expression>
    ...
```

```
if <condition>:
    <expression>
    <expression>
    ...
elif <condition>:
    <expression>
    <expression>
    ...
else:
    <expression>
    <expression>
    ...
```

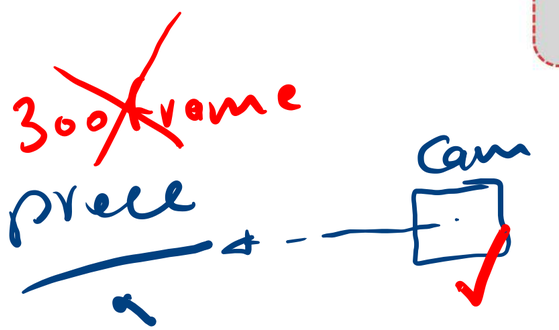
```
age = input('age:')
#convert the string to int
your_age = int(age)

# determine the ticket price
if your_age < 5:
    ticket_price = 5
elif your_age < 16:
    ticket_price = 10
else:
    ticket_price = 18
# show the ticket price
print(ticket_price)
```

if | if...else | if...elif...else | example

while Statement

- <condition> evaluates to a Boolean
- if <condition> is True, do all the steps inside the while code block
- check <condition> again
- repeat until <condition> is False



```
while <condition>:
    <expression>
    <expression>
    ...
```

--->

```
max = 5
counter = 0
while counter < max:
    print(counter)
    counter += 1
```

Ternary Operator

```
if condition:
    value_if_true
else:
    value_if_false
```

Ternary

```
value_if_true if condition else value_if_false
```


```
if age >= 18:
    ticket_price = 20
else:
    ticket_price = 5
```

Ternary

```
ticket_price = 20 if age >= 18 else 5
```

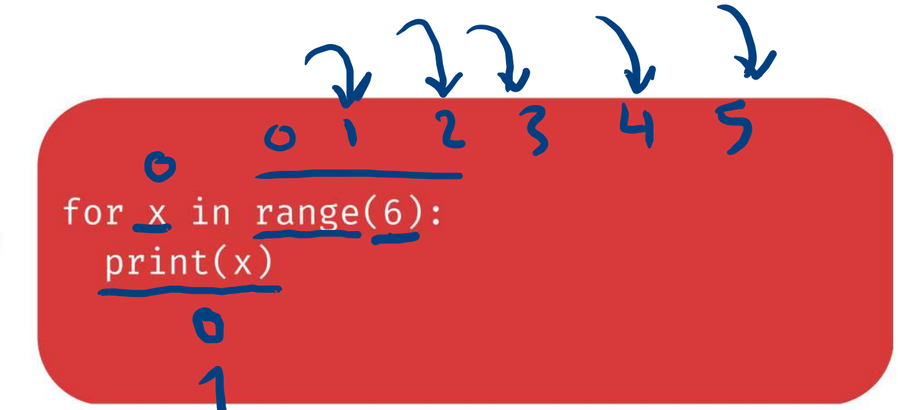
True

- Use the ternary operator to make your code more concise.

for Loops

- Each time through the loop, <variable> takes a value
- First time, <variable> starts at the smallest value
- Next time, <variable> gets the prev value + 1
- Each time through the loop, <variable> takes a value

```
for <variable> in range(<some_num>):
    <expression>
    <expression>
    ...
```



range

range(0, 6)

range(a, b)
start point → end point

range(a, b, c)
c → step

range(0, 6, 2)

- default values are start = 0 and step = 1 and optional
- loop until value is stop - 1

```
>>> mysum = 0
>>> for i in range(7, 10):
        mysum += i
        print(mysum)
```

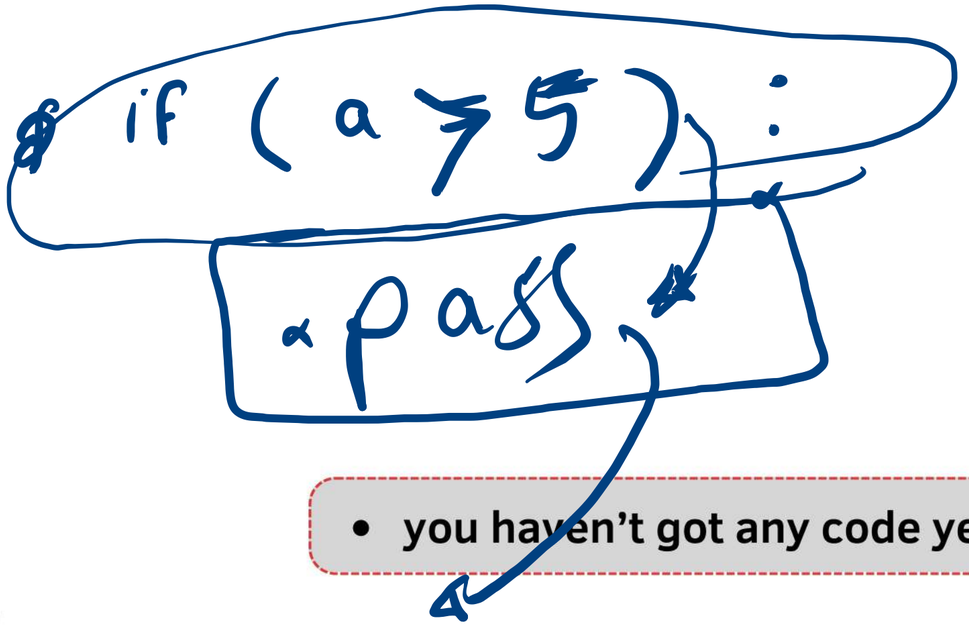
```
>>> mysum = 0
>>> for i in range(5, 11, 2):
        mysum += i
        print(mysum)
```

"break" Statement

- immediately exits whatever loop it is in
- skips remaining expressions in code block
- exits only innermost loop!

```
>>> while <condition_1>:
        while <condition_2>:
            <expression_a>
            break
            <expression_b>
        <expression_c>
```

```
n = 5
while n > 0:
    n-= 1
    if n == 2:
        break
    print(n)
print('Loop ended.')
```



- you haven't got any code yet. But you'll write code for this else clause later.

"pass" Statement

```
counter = 1
max = 10
if counter <= max:
    counter += 1
else:
    pass
```

while vs for

for	while
know number of iterations	unbounded number of iterations
can end early via break	can end early via break
uses a counter	can use a counter but must initialize before loop and increment it inside loop
can rewrite a for loop using a while loop	may not be able to rewrite a while loop using a for loop

Exercises 03

Control Flow & Iteration
900

Control Flow & Iteration

Exercise 01

3! → 3x2x1

- Write a program in Python to display the **Factorial of a number**.

Hints:

- Use **input Method** to get the number!
- output: Factorial of {input number} is {result} !
- output format must be same as the above format, exactly.

```
Example:  input: 9 , output: Factorial of 9 is 362880
```

Control Flow & Iteration

Exercise 02

- Generate random integer number and Write a program to **guess the number.**

Hints:

- Use **randint()** to generate the random number between 1 to 100.
- Use **input Method** to get the number!
- if guessed number is higher than generated number, print "Wrong, Guess the lower number!"
- if guessed number is lower than generated number, print "Wrong, Guess the higher number!"
- if guessed number is equal to the generated number, print "Correct, good job"

```
Example:  Generated Number: 8 , Guessed Number: 26, output: Wrong, Guess the lower number!
```

Control Flow & Iteration

Exercise 03

- Print the following pattern using loops.

```
0 x 0 = 0
1 x 1 = 1
2 x 2 = 4
3 x 3 = 9
4 x 4 = 16
5 x 5 = 25
6 x 6 = 36
7 x 7 = 49
8 x 8 = 64
9 x 9 = 81
10 x 10 = 100
```

Control Flow & Iteration

Exercise 04

- Use for loop to iterate from 0 to n (input number) and print the sum of all numbers.

5

0+1+2+3+4+5

```
Example:  Input Number: 100, output: The sum of all numbers is 5050.
```

Exercise 05 (Search)

- This is a fruit list, ['banana', 'orange', 'mango', 'lemon'] reverse the order using loop.

```
output: [ 'lemon', 'mango', 'orange', 'banana' ]
```

Exercise 06 (Search)

- Find the max element of the [8, 120, 83, 89, 24, 97.5, 201]

```
output: 201
```