

PSPP UNIT 3 ALL PROGRAMS

Simple if statement

The **syntax** of if statement:

if condition:

```
# code to execute if condition is True
```

program:

```
age = 18
```

```
if age >= 18:
```

```
    print("You are old enough to vote")
```

output:

You are old enough to vote

if...else Statement

The **syntax** of if...else statement:

```
if (condition 1):
```

```
    statement 1
```

```
else:
```

```
    statement 2
```

program:

```
age=int(input("Enter your age:"))
```

```
if age >= 18:
```

```
    print("You are old enough to vote")
```

```
else:
```

```
    print("You are not old enough to vote yet")
```

output:

Enter your age:34

You are old enough to vote

if...elif...else statement

The **syntax** of if...elif...else statement:

```
if (condition 1):
```

```
    statement 1
```

```
elif (condition 2):
```

```
    statement 2
```

```
elif (condition 3):
```

```
    statement 3
```

```
else:
```

```
    statement 4
```

program:

```
num=int(input("Enter your number:"))
```

```
if num > 0:
```

```
    print("The number is positive")
```

```
elif num < 0:
```

```
    print("The number is negative")
```

```
else:
```

```
    print("The number is zero")
```

output:

Enter your number:-4

The number is negative

Nested if...else

The syntax of Nested if...else

```
# outer if statement
```

```
if condition1:
```

```
    # statement(s)
```

```
    # inner if-else statement
```

```
        if condition2:
```

```
            # statement(s)
```

```
        else:
```

```
            # alternative statement(s)
```

```
else:
```

```
    # alternative statement(s)
```

program:

```
num = int(input("Enter a number: "))
```

```
if num >= 0:
```

```
    if num == 0:
```

```
        print("You entered zero")
```

```
    else:
```

```
print("You entered a positive number")
```

else:

```
print("You entered a negative number")
```

output:

Enter a number: 23

You entered a positive number

Example 1:

Even-Odd Checker

```
num = int(input("Enter a number: "))
```

```
if num % 2 == 0:
```

```
    print(num, "is even")
```

```
else:
```

```
    print(num, "is odd")
```

output:

Enter a number: 2

2 is even

Example 2:

GREATER THAN -LESS THAN CHECK

```
x=int(input("Enter your number:"))
```

```
if x > 10:
```

```
    print("x is greater than 10")
```

```
elif x < 10:
```

```
print("x is less than 10")
```

else:

```
print("x is equal to 10")
```

output:

Enter your number:45

x is greater than 10

WHILE LOOP

program:

```
count = 1
```

```
while count <= 5:
```

```
    print(count)
```

```
    count += 1
```

output:

1

2

3

4

5

BREAK

program:

```
count = 1
```

```
while count <= 5:
```

```
    print(count)
```

```
    count += 1
```

```
    if count == 4:
```

break

output:

1

2

3

CONTINUE

program:

i = 0

while i < 9:

 i += 1

 if i == 3:

 continue

 print(i)

output:

1

2

4

5

6

7

8

9

PASS

program:

number = 1

```
while number <= 5:
```

```
    pass
```

```
    number += 1
```

```
print("Loop completed")
```

output:

Loop completed

WHILE ELSE

program:

```
count = 0
```

```
while count < 5:
```

```
    print("Count is", count)
```

```
    count += 1
```

```
else:
```

```
    print("Count has reached 5")
```

output;:

Count is 0

Count is 1

Count is 2

Count is 3

Count is 4

Count has reached 5

for loop

program:

```
for i in range (1,10,1):
```

```
    print(i)
```

output:

1

2

3

4

5

6

7

8

9

for loop with a string

program:

```
word = "Hello"
```

```
for letter in word:
```

```
    print(letter)
```

output:

H

e

l

l

o

Fruitful function

program add two numbers:

```
def add(x, y):  
    result = x + y  
    return result
```

```
a = 5
```

```
b = 7
```

```
c = add(a, b)
```

```
print(c)
```

output:

```
12
```

local scope

program:

```
def my_func():  
    x = 5 # local variable  
  
    print(x)
```

```
my_func()
```

Output:

```
5
```

Global scope

program:

```
x = 5 # global variable
```

```
def my_func():
```

```
print(x) # accessing global variable
```

```
my_func()
```

```
print(x) # accessing global variable outside function
```

output:

```
5
```

```
5
```

Function composition

program:

```
def add(x, y):
```

```
    return x + y
```

```
def multiply(x, y):
```

```
    return x * y
```

```
a = 5
```

```
b = 7
```

```
c = 5
```

```
result = multiply(add(a, b), c)
```

```
print(result)
```

output:

```
60
```

Recursion

program:

```
def countdown(n):
```

```
    if n == 0:
```

```
        print("Blastoff!")
```

```
else:
```

```
    print(n)
```

```
    countdown(n-1)
```

```
countdown(5)
```

Output:

5

4

3

2

1

Blastoff!

program 2:

Factorial function using recursion

```
def factorial(n):
```

```
    if n == 0:
```

```
        return 1
```

```
    else:
```

```
        return n * factorial(n-1)
```

```
result = factorial(5)
```

```
print(result)
```

output:

120

string operations:

1:string indexing

```
my_string = "hello"
```

```
print(my_string[0])
```

output:

```
h
```

2:string slicing

```
my_string = "Python"
```

```
print(my_string[1:4])
```

```
print(my_string[:3])
```

```
print(my_string[2:])
```

Output:

```
yth
```

```
Pyt
```

```
thon
```

3:string concatenation

```
my_string = "I like " + "programming"
```

```
print(my_string)
```

output:

```
i like programming
```

4:Repetitions

```
my_string = "python" * 3
```

```
print(my_string)
```

output:

```
pythonpythonpython
```

5:Membership

```
my_string = "hello"
```

```
char = "l"
```

```
if char in my_string:
```

```
    print("Character found!")
```

```
else:
```

```
    print("Character not found.")
```

output:

Character found!

string built in functions and methods:

```
my_string = "hello"
```

```
print(len(my_string)) # Output: 5
```

```
my_string = "hello"
```

```
print(my_string.upper()) # Output: HELLO
```

```
my_string = "HELLO"
```

```
print(my_string.lower()) # Output: hello
```

```
my_string = "hello"
```

```
print(my_string.capitalize()) # Output: Hello
```

```
my_string = " hello "
```

```
print(my_string.strip()) # Output: hello
```

```
my_string = "hello,world"
```

```
print(my_string.split(",")) # Output: ['hello', 'world']
```

```
my_list = ["hello", "world"]  
print(":".join(my_list)) # Output: hello,world
```

```
my_string = "hello world"  
print(my_string.replace("world", "python")) # Output: hello python
```

```
my_string = "hello world"  
print(my_string.count("l")) # Output: 3
```

Array:

program:

```
my_list = ["apple", "banana", "cherry"]  
print(my_list[0]) # Output: apple  
print(my_list[1]) # Output: banana  
print(my_list[2]) # Output: cherry
```

Array built-in methods:

program:

```
my_list = [1, 2, 3]  
my_list.append(4)  
print(my_list) # Output: [1, 2, 3, 4]
```

```
my_list = [1, 2, 4, 5]  
my_list.insert(2, 3)  
print(my_list) # Output: [1, 2, 3, 4, 5]
```

```
my_list = [1, 2, 3, 4, 5]
```

```
my_list.remove(3)
```

```
print(my_list) # Output: [1, 2, 4, 5]
```

```
my_list = [1, 2, 3, 4, 5]
```

```
item = my_list.pop(2)
```

```
print(item) # Output: 3
```

```
print(my_list) # Output: [1, 2, 4, 5]
```

```
my_list = [3, 1, 4, 2, 5]
```

```
my_list.sort()
```

```
print(my_list) # Output: [1, 2, 3, 4, 5]
```

```
my_list = [1, 2, 3, 4, 5]
```

```
my_list.reverse()
```

```
print(my_list) # Output: [5, 4, 3, 2, 1]
```

ILLUSTRATIVE PROGRAMS:

1: Python Program to Find LCM of Two Numbers

```
def gcd(a, b):
```

```
    if b == 0:
```

```
        return a
```

```
    else:
```

```
        return gcd(b, a % b)
```

```
def lcm(a, b):
```

```
    return (a*b)//gcd(a,b)
```

```
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
lcm_num = lcm(num1, num2)
print("LCM of", num1, "and", num2, "is", lcm_num)
```

2:Check Whether a String is Palindrome or Not

```
my_string = input("Enter a string: ")
rev_string = my_string[::-1]
if my_string == rev_string:
    print(my_string, "is a palindrome")
else:
    print(my_string, "is not a palindrome")
```

3:count the number of each vowel in a string

```
my_string = input("Enter a string: ")
my_string = my_string.lower()
vowel_counts = {'a': 0, 'e': 0, 'i': 0, 'o': 0, 'u': 0}
for char in my_string:
    if char in vowel_counts:
        vowel_counts[char] += 1
for vowel, count in vowel_counts.items():
    print("Number of", vowel, "in the string:", count)
```

Binary search:

```
def binary_search(arr, target):
    left = 0
```



```
right = len(arr) - 1
```

```
while left <= right:
```

```
    mid = (left + right) // 2
```

```
    if arr[mid] == target:
```

```
        return mid
```

```
    elif arr[mid] < target:
```

```
        left = mid + 1
```

```
    else:
```

```
        right = mid - 1
```

```
return -1
```

```
arr_str = input("Enter a sorted array of integers, separated by commas: ")
```

```
my_array = [int(i) for i in arr_str.split(",")]
```

```
search_target = int(input("Enter the target value: "))
```

```
result = binary_search(my_array, search_target)
```

```
if result == -1:
```

```
    print("Element not found")
```

```
else:
```

```
    print("Element found at index", result)
```

linear search

```
def linear_search(arr, x):  
    for i in range(len(arr)):  
        if arr[i] == x:  
            return i  
  
    return -1  
  
arr = list(map(int, input("Enter the array elements separated by space: ").split()))  
x = int(input("Enter the element to be searched: "))  
  
result = linear_search(arr, x)  
  
if result == -1:  
    print("Element not found in the array")  
else:  
    print("Element found at index", result)
```