ASSIGNEMENT

BREAK IN PYTHON

The Break statements in python are used to ~~break~~ terminate a loop (either for or while), prematurely.

Once the break statement is encountered within the loop's body, the loop's execution immediately stops and control jumps to the statement after the loop

Continue Statements

Used to skip the remaining code within the current iteration of a loop and jump to the next iteration

Key Differences:

Break statement → exits the entire loop.

Continue statement → skips the remaining code in the current iteration and moves to the next

EXAMPLES OF THE BREAK STATEMENTS

Python Code:

```
fruits = ["apple", "banana", "orange", "grape"]
for fruit in fruits:
    if fruit == "orange":
        print(f"found {fruit}!")
        break
    print("checking next fruit...")
```

(2)

```
for i in range(10):
    if i == 5:
        print("Reached iteration 5, exiting the loop")
        break
    print(i)
```

Output :- 0, 1, 2, 3, 4 [when reached iteration 5, exiting the loop]

(3)

```
while True:
    user_input = input("Enter a number (or 'q' to quit): ")
    print("Existing the loop")
    break
    try:
    number = int(user_input)
    print(f"You entered: {number}")
    except ValueError:
        print("Invalid input, please a number or 'q'.")
```

EXAMPLES OF CONTINUE STATEMENTS

(1)

```
for num in range(10):
    if num % 2 == 0:
        Continue
    print(num)
```

② Python code:

```
numbers = [1, 5, 3, 7, 2, 4]
squared_numbers = []
for num in numbers:
if num == 5  # square only when num is 5
    squared_numbers.append(num * num)
Continue
    squared_numbers.append(num ** 2) # square the other numbers
print(squared_numbers)
# Output: [1, 25, 9, 49, 4, 16].
```

③

```
while True:
print("Menu:")
print("1: option 1")
print("2: option 2")
print("q: Quit")

choice = input("Enter your choice (1, 2, or q): ")
if choice == 'q':
break
print("Invalid choice. please try again.")
```