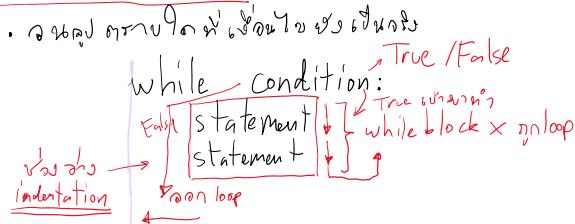


- Agenda:
- 1) Recap
 - 2) List (Part 1)
 - 3) Nested Loops

while structure



Ex:

```

num=0
while num < 4:
    1 num += 1 #num=num+1
    2 print(num)
print("Done")
    
```

Output: 1
2
3
4
Done

```

# loop 1!
0 < 4 ✓
num+=1 → num=1
print(4) → 1
# loop 2:
1 < 4 ✓
num+=1 → num=2
print(2) → 2
↓
    
```

Q19: Given user input - 9999999

```

→ total=0
num = int(input(...))
while num == -9999999:
    total += num
    num = int(input())
print(total)
    
```

```

total=0
while True:
    num = int(input(...))
    if num == -9999999:
        break
    total += num
print(total)
    
```

Input Validation

```

age = int(input(...))
while age < 0:
    print("Age must be > 0")
    age = int(input())
Age < 0 or age > 100 ← * Error handling for age less than 0 or greater than 100
    
```

```

while True:
    age = int(input(...))
    if age >= 0:
        break
    print("Age must be > 0")
if 0 <= age <= 100:
    break
    
```

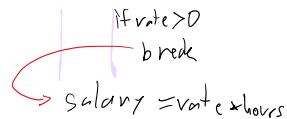
Q20:

```

→ hours = float(input()) ← validate
→ rate = float(input())
salary = rate * hours
print(salary)
    
```

```

while True:
    hours = float(input())
    if hours > 0:
        break
while True:
    rate = float(input())
    if rate > 0:
        break
salary = rate * hours
    
```

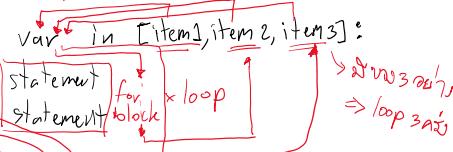



for structure

• \rightarrow Jika \rightarrow max mungkin mungkin dalam

\rightarrow list / range

• Syntax: $\text{for } \underline{\text{var}} \text{ in } [\underline{\text{item1}}, \underline{\text{item2}}, \underline{\text{item3}}]:$



\rightarrow Ex: $\text{for } \underline{\text{num}} \text{ in } [1, 3, 5]:$
 $\quad \text{print}(\underline{\text{num}}, \text{end}='')$ $\rightarrow 1:3:5$

\circledast Ex: \rightarrow n은 loop 변수를 user \rightarrow user가 입력한 수

$n = \text{int}(\text{input}("Enter n:"))$
 $\text{total} = 0$
 $\text{for } \underline{x} \text{ in range}(n):$ \rightarrow seq in 0 to n-1
 $\quad \text{num} = \text{int}(\text{input}("))$ \rightarrow num
 $\quad \text{total} += \text{num}$ \rightarrow total
 $\text{print}(\text{total})$

continue statement

\rightarrow Jika \rightarrow loop 허용 안해 // 97% while, for

\rightarrow while

$\text{num} = 0$
 $\text{while } \underline{\text{num}} < 8:$
 $\quad \text{num} += 1$
 $\quad \text{if } \underline{\text{num}} \% 2 == 0:$ \oplus
 $\quad \quad \text{④ continue}$
 $\quad \quad \text{print}(\underline{\text{num}})$
 $\underline{\text{Output:}}$ $\begin{matrix} 1 \\ 3 \\ 5 \end{matrix}$

\rightarrow for

$\text{for } \underline{\text{num}} \text{ in } [1, 3, 2, 4, 5]:$
 $\quad \text{if } \underline{\text{num}} \% 2 == 0:$ \rightarrow True
 $\quad \quad \text{continue}$
 $\quad \quad \text{print}(\underline{\text{num}})$
 $\underline{\text{Output:}}$ $\begin{matrix} 1 \\ 3 \\ 5 \end{matrix}$

range function

\rightarrow sequence / list \rightarrow max mungkin for!

\rightarrow range (start, stop, step)

\rightarrow ex: $\text{range}(1, 4, 1) \rightarrow [1, 2, 3]$

\rightarrow range (start, stop) # step=1 (default)

\rightarrow ex: $\text{range}(2, 5) \rightarrow [2, 3, 4]$

\rightarrow range (stop) # start=0, step=1

\rightarrow ex: $\text{range}(4) \rightarrow [0, 1, 2, 3]$

List (Part 1)

List \rightarrow data \rightarrow list \rightarrow list \rightarrow list \rightarrow list

type
 \rightarrow max mungkin for loop
 \rightarrow max mungkin index

type
array of items, a way for loop
to access individual items (index)

Syntax: my_list = [item₁, item₂, item₃]

↳ datatype objects
↳ string, int, float, bool, list

Indexing: getting values from array
Syntax: my_list[index]



index from 0

Ex: my_list = [10, 20, 30, 40]
print(my_list[0]) → 10
result = my_list[1] + my_list[3] → 60
print(result) → 60



Error with index 11
e.g. my_list[20] ← Error!
↳ index out of range



negative index

↳ negative index -1 means last element

Ex: my_list = [10, 20, 30, 40]
-4 -3 -2 -1
print(my_list[-1]) → 40
print(my_list[1] + my_list[-2]) → 50
index = 10 + 2 + 1 = 1
print(my_list[index+1]) → 30
1 + 2 = 2 → my_list[2] → 30

Update

* Update on index

Syntax : my_list[index] = new_value

Ex: my_list = [10, 20, 30, 40]
↓ my_list[2] = 50
↓ print(my_list) → [10, 20, 50, 40]
↓ my_list[3] = my_list[2] * 10
↓ print(my_list) → [10, 20, 50, 500]



len function

... len - length of list



len function

↳ von my_list → int my_list

Syntax: `len(my_list)`

Ex `my_list = [1, 3, 5, 7, 9]`
`print(len(my_list))` → 5
`len_list = len(my_list)`
`print(len_list + 1)` → 6

idea: - for every i in list i is index
 - list has index

for loop list

① Output
`my_list = [10, 20, 30, 40]`
`for num in my_list:`
 `print(num)`
`Output: 10`
 `20`
 `30`
 `40`

② Output index
`my_list = [0, 1, 2, 3, 4] → range(5)`
`n = len(my_list)`
`for i in range(n):`
 `print(my_list[i])`
`Output: 0`
 `1`
 `2`
 `3`
 `4`

`n = 4 [0, 1, 2, 3]`
`for i in range(4):`
`#loop 1:`
`i = 0`
`print(num[0])` → 0
`#loop 2:`
`i = 1`
`print(num[1])` → 1
`#loop 3:`
`i = 2`
`print(num[2])` → 2

loop b/w update zu list

* update: `my_list[index] = new_value`

Ex 1 zwei in einer list
`numbers = [1, 2, 3, 4] → [1, 4, 9, 16]`
`for i in range(len(numbers)):`
 `numbers[i] = numbers[i] * 2`
`print(numbers)`
`[1, 4, 9, 16]`

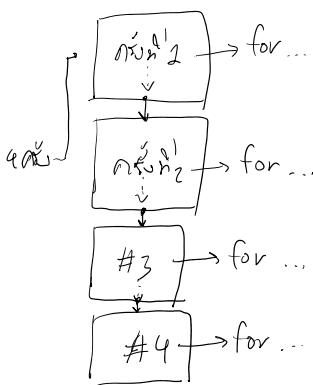
Nested Loop

↳ loop in loop

`for x in range(4):`

`for-block`

b/w for / while



EX

```

for i in range(3):[0,1,2]
    print(i)
    for j in range(2):[0,1]
        print(i,j)
        print("Out of j")
    print("Done")

```

Output:

0	0	1st loop
0	1	
0	Out of j	
1	0	2nd loop
1	1	
1	Out of j	
2	0	3rd loop
2	1	
2	Out of j	

Done ← even Nested loop)

outer 1:
i = 0
print(i) → 0
for j in range(2):
inner 1:
j = 0
print(i,j) → 0,0
inner 2:
j = 1
print(i,j) → 0,1
print("Out of j") → Out of j

outer 2:
i = 1
print(i)
for j in range(2):
inner 1:
j = 0
print(i,j) → 1,0
inner 2:
j = 1
print(i,j) → 1,1
print(...) → Out

1,0

1,1
fj