

# Week 3 Summary

CSCA08

Winter 2022

# String indexing and slicing

- Start index is 0
- Negative indexing (-1 index of last character)
- Slicing:
  - `my_string[start : end : step]`
    - If any of them is skipped the default value replaces:
      - `'csca08'[::-1]`, `'csca08'[1::2]`, `'csca08'[:2:]`
    - Always evaluates to a string
    - We can slice any string:
      - `'hello'[1:4]`
      - `'csca08'[::-1][1::]`

# Booleans

- Only two values: True, False
  - true, false are not defined only True and False are Boolean values
- Boolean expressions:
  - `1 == '1'`, `(3 + 2) != 6.0`, `1 > 0`, `0.0 <= 0`
  - 'cs' in 'csca08'
- Boolean operators:
  - and, or, not
  - not has higher precedence than and, or

# if, elif, else

- Only one of the blocks (the first one whose condition evaluates to True) will be executed.
- Multiple consecutive if statements are not equivalent to if, elif, elif, ..., else.
- We can have nested ifs:

```
if num > 10:
```

```
    if num < 20:
```

```
        ...
```

```
    else:
```

```
        ...
```

```
else:
```

```
    ...
```

# Function always returns

- In Python every function returns a value even if there is no return statement:

```
def my_fun(num: int) -> None:
```

- The function execution end (and we jump out of the function) as soon as we return a value.

```
def my_fun(num: int) -> str:  
    if num > 10:  
        return 'greater than ten'
```

The above function can return None!

# print vs return

- Print only displays but does not return any value. You can not assign the value printed on the screen to any variable.
- The following function returns None:

```
def my_fun(num: int) -> None:  
    print(num)
```

Try calling it this way:

```
>>> a = my_fun(10)
```

It will print 10 on the screen but the value of a is None!