

CSC 101

Today

- Conditionals

- Lists Intro.

- Quiz 3

- Assignment 2

```
def max_of_three(x: float, y: float, z: float) →  
    float:
```

if $x > y$ and $x > z$:

return x

$\leftarrow x > y$ and
 $x > z$

elif $y > z$:

return y

before check

$\neg(x > y \text{ and } x > z)$

$\rightarrow \neg(x > y) \text{ or }$
 $\neg(x > z)$

else:

return z

$(x \leq y \text{ or } x \leq z)$
and $y > z$

$x \leq y \text{ or }$
 $x \leq z$

$x \leq y \text{ and } y > z$ or
 $x \leq z \text{ and } y > z$

$(x \leq y \text{ or } x \leq z)$
and
 $\neg(y > z)$
 $y \leq z$

if $a < b$:

if $c < d$:



else:



if $a < b$:



else:

if $c < d$:



else:



if $a < b$:



elif $e < d$:



else:



DeMorgan's Laws

$$\text{not } (\text{A and B}) = \text{not A or not B}$$

$$\begin{array}{ccc} \uparrow & & \uparrow \\ x < y & & x < z \end{array}$$

$$\text{not } (\text{A or B}) = \text{not A and not B}$$

A	B	A or B	not (A or B)	not A and not B
T	T	T	F	F
T	F	T	F	F
F	T	T	F	F
F	F	F	T	T

not A	not B
F	F
F	T
T	F
T	T

Prices

age

under 10

standard

13

matinee (1600)

12

adult

20

19

over 55

18

16

```
def price (age: int , time: int ) → float :
```

```
    if age < 10 : # child  
        return child_price (time)
```

```
    elif age > 55 : # senior  
        return senior_price (time)
```

```
    else :  
        return adult_price (time) # adult
```

```
def child_price (time: int ) → float :
```

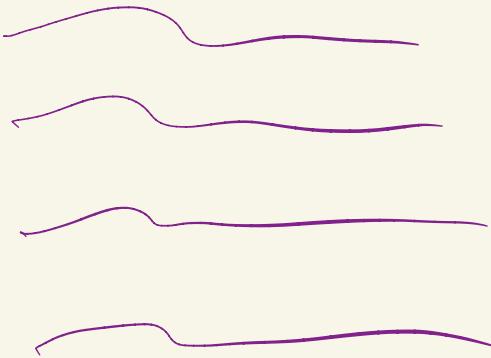
```
    if time < 1600 :  
        return 12
```

```
    else : return 13
```

A	B	A or B	A ^(xor) ^ B
T	T	T	F
T	F	T	T
F	T	T	T
F	F	F	F

play list

song¹ = Song (artist, ---)
song² = Song (ar - - -



⋮

List

- sequential data



3 4 7

list = [3, 4, 7]
= $\begin{matrix} 0 & 1 & 2 \\ 3 & 4 & 7 \end{matrix}$

0-based
indexing

$$x = \text{list}[0] + 4$$



Known Values

list \rightarrow

3	4	7
---	---	---

$$x \boxed{7}$$

$$y = \underbrace{\text{list}[2]}_7 + 4$$

Steps

$$x = \text{list}[0] + 4$$

$$x = \boxed{3}[0] + 4$$

1st

$$x = 3 + 4$$

$$x = 7$$

$$x = ([3, 4, 7])[0] + 4$$

list is indexed
starting @ 0
ending @ length -1

points = [Point(⁰(2, 3), Point(¹(4, 7))]

~~songs~~ playlist = [Song(⁰...), Song(¹...), Song(²...)]