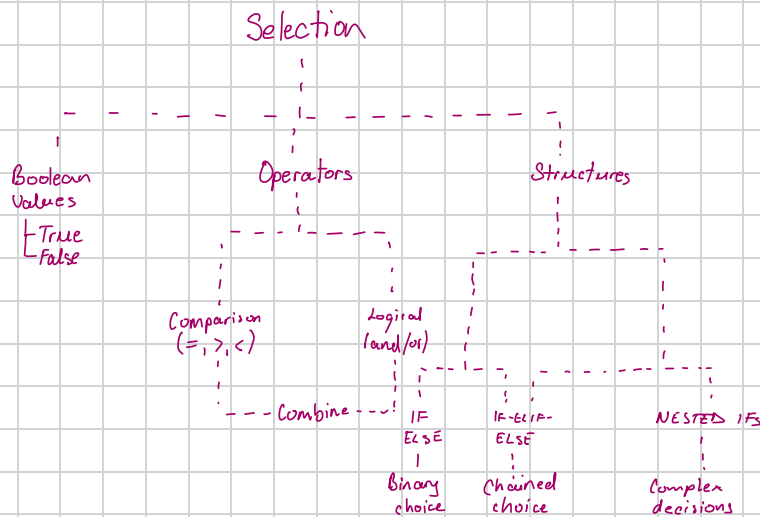


CHAPTER 7



71 Boolean Values: True or False

The Basics:

Only 2 values: True or False (like light switches: on/off)

Boolean expressions = questions that give True/False answers

Examples: $5 > 3 \rightarrow \text{True}$, $10 == 7 \rightarrow \text{False}$

Comparison Operators:

$==$ equals, $!=$ not equals

$>$ greater than, $<$ less than

$>=$ greater or equal, $<=$ less or equal

72 Logical Operators: Combining Conditions

Three Magic Words:

and - BOTH must be True

(age ≥ 18) and (hasLicense == True) \rightarrow can drive

or - AT LEAST ONE must be True

(isWeekend) or (isHoliday) \rightarrow can sleep in

not - FLIPS the answer

not(isRaining) \rightarrow it's sunny!

💡 Logical Opposites (De Morgan's Laws):

not(A and B) = (not A) or (not B)

not(A or B) = (not A) and (not B)

73 Precedence of Operators: Order Matters!

Remember PEMDAS + Comparisons + Logicals:

Parentheses ()

Exponents **

Multiplication/Division * /

Addition/Subtraction + -

Comparisons < > == !=

not

and

or

Tip: When in doubt, use parentheses!

74 Binary Selection: The Classic IF-ELSE

if condition:

do this if True

else:

do this if False

Think: Two paths, program picks one based on condition

Like choosing: "If hungry \rightarrow eat pizza, else \rightarrow keep studying"

75 Unary Selection: Just IF (No ELSE)

if condition:

do this if True

continue either way

Use when: You only need to do something special

in one case

Like: "If it's your birthday \rightarrow sing happy birthday" (otherwise just continue normal day)

76 Nested Conditionals: IF inside IF

if outside_condition:

if inside_condition:

both must be True

else:

outer True, inner False

else:

outer False

Think: Russian nesting dolls - decisions within decisions

Example: "If have money \rightarrow if store open \rightarrow buy snacks"

77 Chained Conditionals: IF-ELIF-ELSE

if condition1:

first option

elif condition2:

second option

elif condition3:

third option

else:

default option

Think: Multiple choice test - picks FIRST True

condition

Like grading: "If score $\geq 90 \rightarrow$ A, elif ≥ 80

\rightarrow B, elif $\geq 70 \rightarrow$ C, else \rightarrow F"

⚠ Important: Stops at first True! Order

matters!

78 Boolean Functions: Reusable Decision Makers

def is_passing(score):

return score ≥ 60 # returns True or False

Benefits:

Clean, readable code

Reuse logic everywhere

Easy to test!

Unit Testing:

Test True cases

Test False cases

Test edge cases (boundaries)