

Conditionals

CSCI 1030U - Intro to Computer Science
@IntroCS

Randy J. Fortier
@randy_fortier

Outline

- Conditionals
 - if statements
 - if/else statements
 - if/elif/else statements
 - Conditional expressions

Conditionals

Conditionals



Conditionals

- A conditional is a programming language statement that optionally executes a block of code
 - Conditionals evaluate a condition, which is an expression of Boolean type
 - If the condition is true, the code block is executed
 - If the condition is false, the code block is ignored

Conditionals - If

- An *if* statement is the most common conditional statement
 - An if statement evaluates a condition, which is an expression of Boolean type
 - If the condition is true, the code block is executed
 - If the condition is false, the code block is ignored

Python:

```
if value < 10:  
    print('Small value')
```

C++:

```
if (value < 10) {  
    cout << "Small value";  
}
```

Conditionals - If

```
pokemon_hp = -2  
pokemon_name = 'Tepig'  
if pokemon_hp < 0:  
    print(f'{pokemon_name} has fainted!')  
print('All done')
```

0

...

Conditionals - If

```
pokemon_hp = 5  
pokemon_name = 'Tepig'  
if pokemon_hp < 0:  
    print(f'{pokemon_name} has fainted!')  
print('All done')
```

0

...

Conditionals - If/Else

- An *if* statement can also have an *else* clause
 - An if statement evaluates a condition, which is an expression of Boolean type
 - If the condition is true, the code block in the if clause is executed
 - If the condition is false, the code block in the else clause is executed

Python:

```
if value < 10:  
    print('Small')  
else:  
    print('Not small')
```

C++:

```
if (value < 10) {  
    cout << "Small";  
} else {  
    cout << "Not small";  
}
```

Conditionals - If/Else

```
pokemon_type = 'Fire'
move_type = 'Fire'
if pokemon_type == move_type:
    move_damage = 30
else:
    move_damage = 20
print('All done')
```

0

...

Conditionals - If/Else

```
pokemon_type = 'Fire'
move_type = 'Fighting'
if pokemon_type == move_type:
    move_damage = 30
else:
    move_damage = 20
print('All done')
```

0

...

Conditionals - Mutually Exclusive

```
pokemon_type = 'Fire'
move_type = 'Fighting'
if pokemon_type == move_type:
    move_damage = 30
else:
    move_damage = 20
print('All done')
```

pokemon_type	str	
move_type	str	

0	...
14	'Fire'
	...
29	'Fighting'
	...

Conditionals - If/Elif/Else

- An *if* statement can also have an *elif* or *else if* clause
 - An *elif* clause is an additional condition
 - If the *if* condition is false, the *elif* condition(s) are then checked (in order)
 - Finally, the *else* code is executed if none of the previous conditions are true

Python:

```
if value < 10:  
    print('Small')  
elif value < 20:  
    print('Medium')  
else:  
    print('Large')
```

C++:

```
if (value < 10) {  
    cout << "Small";  
} else if (value < 20) {  
    cout << "Medium";  
} else {  
    cout << "Large";  
}
```

Conditionals - If/Elif/Else

```
opponent_type = 'Fire'
if opponent_type == 'Fire':
    move_damage = 10
elif opponent_type == 'Water':
    move_damage = 30
else:
    move_damage = 20
print('All done')
```

0

...

Conditionals - If/Elif/Else

```
opponent_type = 'Water'
if opponent_type == 'Fire':
    move_damage = 10
elif opponent_type == 'Water':
    move_damage = 30
else:
    move_damage = 20
print('All done')
```

0

...

Conditionals - If/Elif/Else

```
opponent_type = 'Grass'
if opponent_type == 'Fire':
    move_damage = 10
elif opponent_type == 'Water':
    move_damage = 30
else:
    move_damage = 20
print('All done')
```

0

...

A Note on Testing

- When testing a program, it is important to test all code paths
 - In a conditional, there are at least two code paths:
 - If the condition is true (`if`)
 - If the condition is false (`else`)
 - Each `elif` condition is another code path
- The ideal is to use different inputs to test each of these code paths
 - This is called 100% code coverage

Demo - Conditionals

Programming Exercise 02b.1

- Write a program that asks the user for two numbers, and outputs the message `Both numbers are even` only if both numbers are even

Programming Exercise 02b.2

- Write a program that checks the value of the `health_points` variable
 - If `health_points` is less than or equal to 0, then set the variable `is_dead` to `True`
 - If `health_points` is greater than 0, then set the variable `is_dead` to `False`

Programming Challenge 02b.1

- Write a program that asks the user for a single mark (out of 100), and outputs the letter grade that corresponds to that mark
- Use the following ranges:
 - 0–49: F
 - 50–59: D
 - 60–69: C
 - 70–79: B
 - 80–100: A

Hackers' Corner

- Conditional expressions
 - Also called the ternary operator, this is a shortcut operator for choosing between two values:

Python:

```
maximum = x if x > y else y
```

C++:

```
int maximum = x > y ? x : y;
```

Wrap-up

- Conditionals
 - if statements
 - if/else statements
 - if/elif/else statements
 - Conditional expressions

Coming Up

- Loops
 - For loops
 - While loops