

if-then-else

Conditional Control Statement

Example Setup

In VSCode:

1. Start the Development Server
 - View Terminal
 - `npm run pull`
 - `npm start`
2. Open the File Explorer Pane
 - Right click on the src folder
 - Select "New folder"
 - Name it: **x-if-then-else**
 - Right click on the x-if-then-else folder
 - Select "New file"
 - Name it: **if-then-else-app.ts**
3. In `if-then-else-app.ts`, add the code to the right.

```
import { print, promptNumber } from "intros";
export let main = async () => {
    print("I'm thinking of a number...");
    let guess = await promptNumber("Guess:");
    print("Your guess is...");
    if (guess === 42) {
        print("Correct!");
    }
    print("Game Over");
};
main();
```

if-then Statements

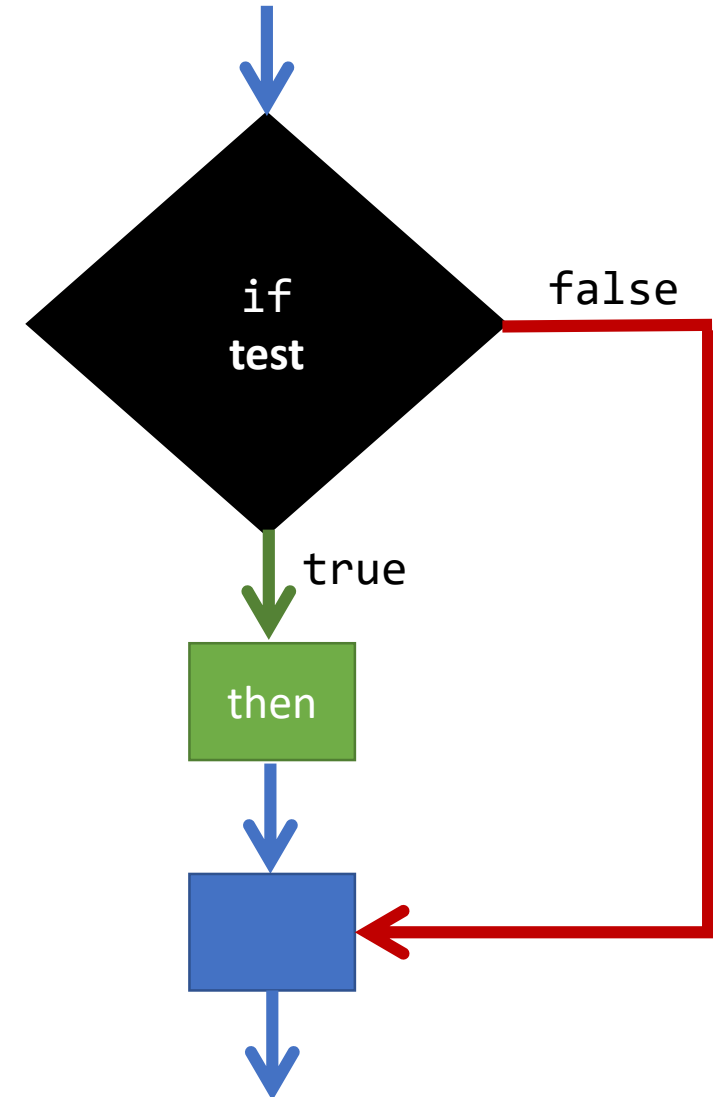
- General form of an **if-then** statement:

```
if (<"test" - boolean expression>) {  
    <then block - runs when test is true>  
}
```

- **if-then** is a **control statement**
 - It can be written *anywhere* you can write any other statement
 - It is like a conditional phrase at the beginning of a sentence (and does not end in a semi-colon)
- The **test** in the parenthesis must be a **boolean** expression
- Statements in the "**then block**" curly brace will run ***if*** the **test** evaluates to **true**. ***Else***, the processor *jumps* over the then block.

if-then Statements

- In a flow chart ("*control flow*") we draw an *if-then* statement as a diamond.
- It will have two arrows coming out. We label these arrows for the two cases:
 - **true** will point to code in the *then* block
 - **false** will point to code following the *then* block



```
print("Your guess is...");
```

`if (guess === 42)`

false

true

```
{  
  print("Correct! ");  
}
```

```
print("Game Over");
```

How do we follow a different path when the test condition is **false**?

Example - Add an **else** clause

Add an **else** clause like the one to the right.

Try playing your game again and entering a correct guess as well as an incorrect guess.

```
if (guess === 42) {  
    print("Correct!");  
} else {  
    print("Nope!");  
}
```

if-then-else Statements

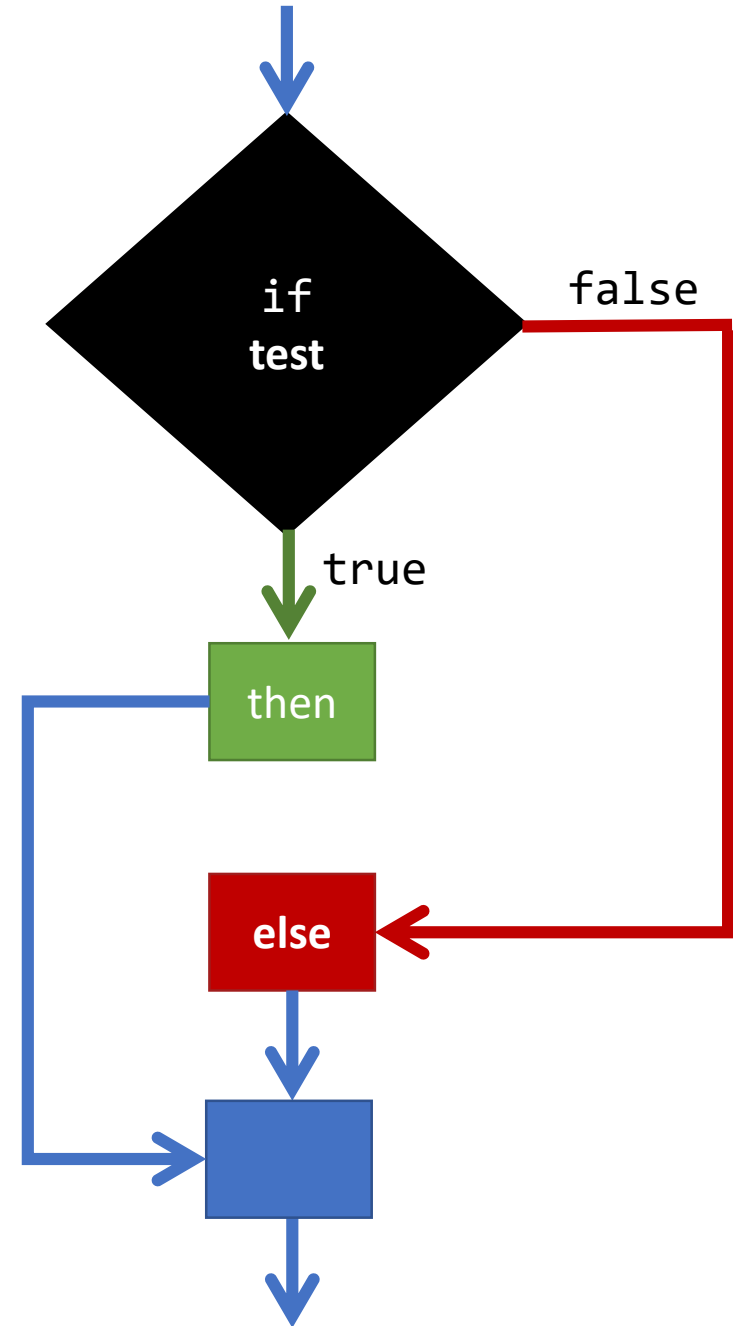
- General form of an **if-then-else** statement:

```
if (<"test" - boolean expression>) {  
    <then block - runs when test is true>  
} else {  
    <else block - runs when test is false>  
}
```

- Works the same as an **if-then** statement, however, when the **test** expression evaluates to **false** the statements within the else block will run.
- Once either block completes, the processor resumes at the line following the else block.

if-then-else Statements

- Notice, like the `if-then` statement, the then block runs **only** when the test condition is **true**
- Unlike the `if-then` statement, the else block runs **only** when the test condition is **false**
- After *either* the then-block or else-block complete, they both continue to the same next step



Nesting if-then-else statements *within* if-then-else statements

- The *then* and *else* blocks may contain one or more **statements**...
- ...but isn't *if-then* a statement?
 - Yes!
- You can write further *if-then* statements inside of **then** or **else** blocks and the *same* rules apply.

Example - Add a nested `if-then-else` statement

Add the nested if-then-else statement to the right inside of the else block.

Your game should now indicate if the guess was too high or too low!

```
if (guess === 42) {  
    print("Yep!");  
} else {  
    if (guess > 42) {  
        print("Too high!");  
    } else {  
        print("Too low!");  
    }  
}
```