
1. Basic if Statement

An **if statement** runs **only if the condition is true**.

```
var x = prompt("Where does the Pope live?");
```

```
if (x === "Vatican") {  
    alert("Correct!");  
}
```

- ✦ If the user enters "Vatican", the **alert appears**.
- ✦ If they enter anything else, **nothing happens**.

2. Using `else` for Two Outcomes

You often want **something to happen in both cases** (right or wrong answer).

```
var x = prompt("Where does the Pope live?");
```

```
if (x === "Vatican") {  
    alert("Correct!");  
} else {  
    alert("Wrong answer");  
}
```

✦ Here:

✓ If the answer is "Vatican", it shows **"Correct!"**

✓ Otherwise, it shows **"Wrong answer"**

- ✦ This is **shorter** and **easier to read** than writing two separate if statements.

3. Running Multiple Statements in `else`

You can have **more than one action** inside the if or else block.

```
var x = prompt("Where does the Pope live?");  
var score = 10;
```

```
if (x === "Vatican") {  
  alert("Correct!");  
} else {  
  score--; // Decrease score by 1  
  alert("Incorrect. Your score is now " + score);  
}
```

- ✓ If the user is wrong, **their score decreases**.
 - ✓ You can also **update variables inside if/else**.
-

4. else if for Multiple Conditions

When you want to check **more than one possible correct answer**, use else if.

```
var x = prompt("Where does the Pope live?");
```

```
if (x === "Vatican") {  
  alert("Correct!");  
} else if (x === "Rome") {  
  alert("Incorrect, but close!");  
} else {  
  alert("Incorrect");  
}
```

- ✓ If the user enters **"Vatican"**, it shows **"Correct!"**
- ✓ If they enter **"Rome"**, it shows **"Incorrect, but close!"**
- ✓ For **any other answer**, it shows **"Incorrect"**

⚡ JavaScript **stops checking** as soon as it finds a true condition.

5. Order Matters in else if

Since JavaScript stops at the **first true condition**, order is important.

```
var x = prompt("Where does the Pope live?");
```

```
if (x !== "Vatican") {  
  alert("Wrong answer");  
} else if (x === "Rome") {  
  alert("Incorrect, but close!");  
} else {
```

```
    alert("Correct!");  
}
```

✗ This **won't work correctly** because:

- The **first condition** (`x !== "Vatican"`) is true for "Rome", so JavaScript runs that block and **stops checking further**.
- The "Incorrect, but close!" alert **never runs**.

✓ **Correct way:** Always **put the most specific condition first**.

6. Nested if Statements

You can put an if statement **inside another**.

```
var x = prompt("Where does the Pope live?");  
var y = prompt("Are you sure? (yes/no)");
```

```
if (x === "Vatican") {  
    if (y === "yes") {  
        alert("Great! You are confident.");  
    } else {  
        alert("You need more confidence!");  
    }  
} else {  
    alert("Wrong answer");  
}
```

✓ If the answer is "Vatican", it asks if the user is **sure**.

✓ If "yes", it says **"Great! You are confident."**

✓ Otherwise, it says **"You need more confidence!"**

7. The Shorter if-else (Ternary Operator)

For **very simple if-else cases**, you can use this shorter version:

```
var x = prompt("Where does the Pope live?");  
alert(x === "Vatican" ? "Correct!" : "Wrong answer");
```

✦ How it works?

- ✓ If `x === "Vatican"`, it shows "Correct!".
- ✓ Otherwise, it shows "Wrong answer".

⚡ **Use this only for short cases** to keep code readable.

Summary: if-else **Best Practices**

- ✓ **Use `else` to handle the "wrong" case** instead of writing two if statements.
- ✓ **Use `else if` for multiple conditions** (e.g., "Vatican", "Rome", etc.).
- ✓ **Order matters in `else if`** – more specific conditions should come first.
- ✓ **Use nested if only when necessary**, to keep code readable.
- ✓ **For very short conditions, use the ternary operator (`? :`).**