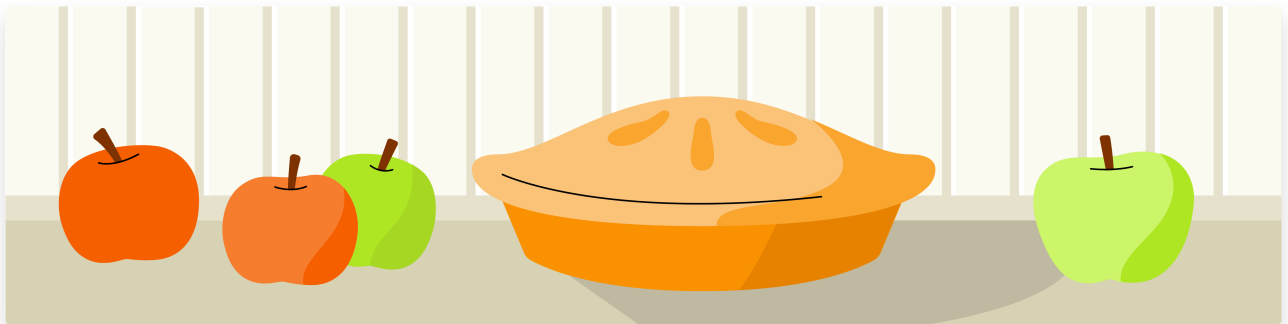


Conditionals

Textbook

Conditionals



You can smell the warm baked pie sitting on the stove cooling down. Your mouth is watering at the idea of having a slice. But you've already been told you can't have dessert until you eat your dinner. This is an example of **conditionals** in real life. It's when things happen only under certain conditions. Just like in life, conditionals in programming help control what happens and when!

What Are Conditionals?

Conditionals are pieces of code that run only *if* specific conditions are true. They act like "if-then" rules:

- **If** something happens, **then** do this.
- **Otherwise**, do something else.

For example:

- **If** it's raining outside, **then** take an umbrella.
- **If** your score in a game reaches 50, **then** you unlock a new level.

Conditionals in Programming

Conditionals are important in programming because they allow you to control your code. They can make programs more dynamic and interactive.

- In a game: **If** a player collects all the keys, **then** unlock the treasure chest.
- In a weather app: **If** the temperature is below 32°F, **then** display a snowflake icon.

Parts of a Conditional

1. **Condition:** The rule or test that must be true.
2. **Action:** What happens if the condition is met.

Conditionals Video

The screenshot shows a Skill Struck video player interface. On the left, a sidebar displays the lesson title "Java Conditionals" and a "Vocabulary" section. The main video area shows a balance scale with two green apples on the left and three orange oranges on the right. Below the scale, text explains that a condition compares values using a comparison operator. The video player includes a progress bar at the bottom showing 0:00 / 4:41 and a "Start Checkpoint" button. On the right, a code editor displays the following Java code for a class named Cereal:


```
1 public class Cereal {
2
3
4 public static void main(String
5 [] args) {
6
7
8
9 }
10
11 }
```

Below the code editor, a console window shows the output of the program, which is the number -17.

Coding Conditionals




For this puzzle, you will want to put multiple commands inside your IF statement.

Blocks	Syntax
	<pre> 1 if (isClearLeft()) { 2 turnLeft(); 3 moveForward(); 4 } 5 </pre>

The `isClearLeft` condition allows the character to check if the space to its left is clear, then move to it.

You can also write whatever code you want before or after your IF statement.

Blocks	Syntax
	<pre> 1 if (isClearLeft()) { 2 turnLeft(); 3 moveForward(); 4 } 5 turnRight(); 6 moveForward(); 7 if (isClearAhead()) { 8 moveForward(); 9 } </pre>

JavaScript Coding Tips

Coding a conditional in [JavaScript](#) looks like this:

```

1 if () {
2
3 }

```

Your condition goes between the `()` parenthesis.

What you want your character to do goes between the `{ }` braces.

For example, if you want your character to move forward **IF** it is clear ahead, it would look like this:

```
1 if (isClearAhead()) {  
2   moveForward();  
3 }
```

This will move the character forward only IF the path is clear ahead.

Critical Thinking Questions

1. Can you think of a time when you made a decision based on specific conditions, like "If I finish my homework, then I can play outside"? How does this compare to conditionals in programming?
2. Why do you think conditionals are important in games, apps, and other software? Can you give an example of a program you use that might include conditionals?

Questions (5)

1. What is a conditional in programming?

MULTIPLE CHOICE

Choose the correct answer:

- A. A box that holds information
- B. Clicking an icon
- C. A sequence of code that repeats
- D. Something that will happen only IF something else happens.

2. True or False: Conditionals are useful in programming because they can help you time when code will run

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

3. When would you use a conditional in a game program?

MULTIPLE CHOICE

Choose the correct answer:

- A. To add more colors to the game
- B. To change the character's name
- C. To control when certain actions happen based on conditions

4. What does a conditional statement in JavaScript look like?

MULTIPLE CHOICE

Choose the correct answer:

- A. `if [] condition{ }`
- B. `if () { }`
- C. `if < > { }`

5. If you want your character to move left only if the space to its left is clear, what would the conditional look like?

MULTIPLE CHOICE

Choose the correct answer:

- A. `if (isClearLeft()) { moveLeft(); }`
- B. `if (moveLeft()) { isClearLeft(); }`
- C. `if (moveLeft() { isClearLeft(); })`

Games (2)

1. Conditionals Typing Game

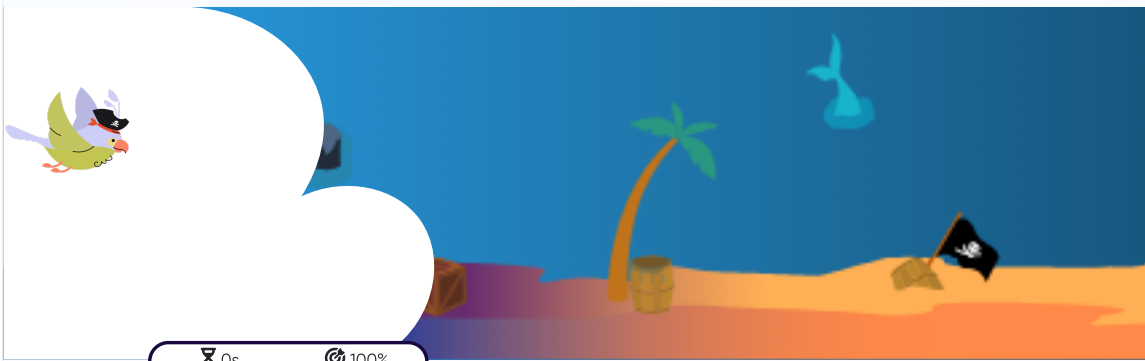
Full Screen

Audio

Instructions

Restart

Pause



0s 100%

A conditional is a piece c

2. Conditionals Memory Game

Full Screen

Audio

Instructions

Answer Key

Pause

Flips: 0

1

2

3

4

5

6

7




8



9


10


Puzzles (14)


1. Conditionals #1


Instructions 
Answer Key 
</> 


Start 
Start 

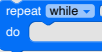












































Start 
Start 

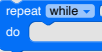












































Start 
Start 

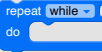












































Start 
Start 

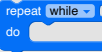












































Start 
Start 

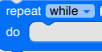

































2. Conditionals #2

Instructions
Answer Key
</>

Start

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

isClearRight

▶

↗

👤

🏠

Minimum Conditionals
1

3. Conditionals #3

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end

not

isClearAhead

isClearLeft

isClearRight

Minimum
Conditionals
1

4. Conditionals #4

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end

not

isClearAhead

isClearLeft

isClearRight

Minimum
Conditionals
1

5. Conditionals #5

Instructions
Answer Key

Start
moveForward
turnRight
turnLeft
turnAround
repeat 10 times
do
repeat while
do
if
do
end
not
isClearAhead
isClearLeft
isClearRight

Start

Minimum Conditionals
1

6. Conditionals #6

Instructions
Answer Key

Start
moveForward
turnRight
turnLeft
turnAround
repeat 10 times
do
repeat while
do
if
do
end
not
isClearAhead
isClearLeft
isClearRight

Start

Minimum Conditionals
1

7. Conditionals #7

Instructions
Answer Key
</>

Start

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

isClearRight

▶

⬆

⬆

⬆

Minimum
Conditionals
1

8. Conditionals #8

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

isClearRight

Start

▶

↗

👤


🔔


Minimum



Conditionals

1

9. Conditionals #9

Instructions 

Answer Key 

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end




not





isClearAhead

isClearLeft

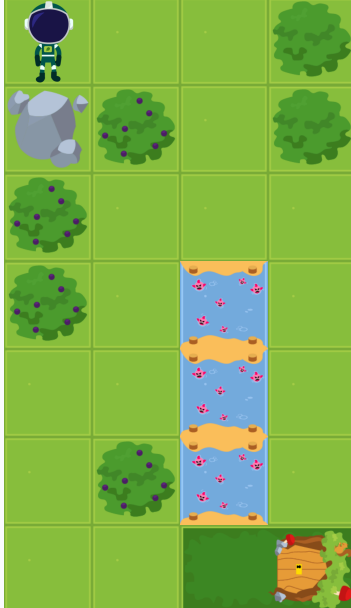
isClearRight

Start














Minimum
Conditionals
1



10. Conditionals #10

Instructions 

Answer Key 

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end




not





isClearAhead

isClearLeft


isClearRight

Start

Minimum
Conditionals
1



11. Conditionals #11

Instructions
Answer Key

Start

```

moveForward
turnRight
turnLeft
turnAround
pullLever

repeat 10 times
do

repeat while
do

if
do

and

not

isClearAhead

isClearLeft
        
```

Start

Minimum Conditionals
1

12. Conditionals #12

Instructions
Answer Key
</>

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times
do

repeat while
do

if
do

and

not

isClearAhead


isClearLeft


Start

▶


Minimum
Conditionals
1

13. Conditionals #13

Instructions 

Answer Key 

</>



Start

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if




do





and


not

isClearAhead

isClearLeft















Minimum
Conditionals
1

14. Conditionals #14

Instructions 

Answer Key 

</>



Start

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if




do





and


not

isClearAhead

isClearLeft



Minimum
Conditionals
1

Answer Keys & Solutions

Questions

1. What is a conditional in programming?

MULTIPLE CHOICE

Correct Answer:

- A. A box that holds information ✗ Incorrect
- B. Clicking an icon ✗ Incorrect
- C. A sequence of code that repeats ✗ Incorrect
- D. Something that will happen only IF something else happens. ✓ Correct

Explanation:

A conditional states, "If this happens, then this will happen"

2. True or False: Conditionals are useful in programming because they can help you time when code will run

MULTIPLE CHOICE

Correct Answer:

- A. True ✓ Correct
- B. False ✗ Incorrect

Explanation:

Conditionals let you decide when things take place.

3. When would you use a conditional in a game program?

MULTIPLE CHOICE

Correct Answer:

- A. To add more colors to the game ✗ Incorrect
- B. To change the character's name ✗ Incorrect

C. To control when certain actions happen based on conditions

✓ Correct

Explanation:

Think about the example where a player must complete tasks to unlock access to the next level.

4. What does a conditional statement in JavaScript look like?

MULTIPLE CHOICE

Correct Answer:

A. `if [] condition{}`

✗ Incorrect

B. `if () { }`

✓ Correct

C. `if < > { }`

✗ Incorrect

Explanation:

Recall the format shown in the JavaScript coding tip where the condition is inside parentheses and actions are inside curly braces.

5. If you want your character to move left only if the space to its left is clear, what would the conditional look like?

MULTIPLE CHOICE

Correct Answer:

A. `if (isClearLeft()) { moveLeft(); }`

✓ Correct

B. `if (moveLeft()) { isClearLeft(); }`

✗ Incorrect

C. `if (moveLeft() { isClearLeft(); })`

✗ Incorrect

Explanation:

Remember the structure: the condition goes inside the parentheses, and the action (moving left) goes inside the curly braces.



Games

1. Conditionals Typing Game

Typing game - no answer key needed. Students practice typing the provided content.

2. Conditionals Memory Game

Memory Game Pairs:

1.  An orange code block that says "If...do" ↔  An orange code block that says "If...do"
2.  A dark code block that says "isClearAhead" ↔  A dark code block that says "isClearAhead"
3.  A dark code block that says "isClearLeft" ↔  A dark code block that says "isClearLeft"
4.  A dark code block that says "isClearRight" ↔  A dark code block that says "isClearRight"
5.  A dark code block that says "isClearBehind" ↔  A dark code block that says "isClearBehind"

Students must find all matching pairs by flipping cards and remembering their positions.

Puzzles

1. Conditionals #1

Solution:

```
1 if (isClearRight()) {  
2   turnRight();  
3 moveForward();  
4 moveForward();  
5 moveForward();  
6 moveForward();  
7 }
```

2. Conditionals #2

Solution:

```
1 if (isClearAhead()) {  
2   moveForward();  
3 }  
4 moveForward();  
5 moveForward();  
6 turnLeft();  
7 moveForward();  
8 moveForward();
```

3. Conditionals #3

Solution:

```
1 if (isClearAhead()) {
2   moveForward();
3   moveForward();
4   moveForward();
5   moveForward();
6 }
7 turnLeft();
8 moveForward();
9 moveForward();
10 turnLeft();
11 if (isClearAhead()) {
12   moveForward();
13   moveForward();
14   moveForward();
15   moveForward();
16 }
```

4. Conditionals #4**Solution:**

```
1 if (isClearAhead()) {
2   moveForward();
3 }
4 moveForward();
5 turnLeft();
6 moveForward();
7 moveForward();
8 turnLeft();
9 moveForward();
10 moveForward();
11 turnRight();
12 moveForward();
13 moveForward();
14 turnRight();
15 moveForward();
16 moveForward();
17 moveForward();
18 moveForward();
19 turnRight();
20 moveForward();
21 moveForward();
22 moveForward();
```

5. Conditionals #5**Solution:**

```
1 if (isClearRight()) {
2   turnRight();
3   moveForward();
4 }
5 turnRight();
```

```
6 moveForward();
7 moveForward();
8 moveForward();
9 moveForward();
10 turnLeft();
11 moveForward();
12 moveForward();
```

6. Conditionals #6

Solution:

```
1 if (isClearAhead()) {
2     moveForward();
3     turnLeft();
4 }
5 moveForward();
6 moveForward();
7 turnRight();
8 moveForward();
9 moveForward();
10 turnRight();
11 moveForward();
12 turnLeft();
13 moveForward();
14 moveForward();
15 turnLeft();
16 moveForward();
17 moveForward();
18 turnLeft();
19 moveForward();
20 turnRight();
21 moveForward();
22 moveForward();
```

7. Conditionals #7

Solution:

```
1 if (isClearBehind()) {
2     turnAround();
3     moveForward();
4 }
5 turnLeft();
6 moveForward();
7 turnRight();
8 moveForward();
9 moveForward();
10 turnLeft();
11 moveForward();
12 turnRight();
13 moveForward();
14 turnLeft();
15 moveForward();
```

8. Conditionals #8

Solution:

```
1 if (isClearAhead()) {
2     moveForward();
3     moveForward();
4     moveForward();
5 }
6 turnLeft();
7 moveForward();
8 turnRight();
9 moveForward();
10 turnLeft();
11 moveForward();
12 moveForward();
13 turnRight();
14 moveForward();
15 moveForward();
16 turnLeft();
17 moveForward();
18 moveForward();
```

9. Conditionals #9

Solution:

```
1 if (isClearLeft()) {
2     turnLeft();
3     moveForward();
4     moveForward();
5 }
6 if (isClearRight()) {
7     turnRight();
8     moveForward();
9     moveForward();
10 }
11 turnRight();
12 moveForward();
13 if (isClearLeft()) {
14     turnLeft();
15     moveForward();
16     moveForward();
17 }
18 turnRight();
19 moveForward();
20 if (isClearLeft()) {
21     turnLeft();
22     moveForward();
23     moveForward();
24 }
25 if (isClearLeft()) {
26     turnLeft();
27     moveForward();
28     moveForward();
```

```
29 }
```

10. Conditionals #10

Solution:

```
1  if (isClearRight()) {
2    turnRight();
3    moveForward();
4    moveForward();
5  }
6  turnRight();
7  moveForward();
8  turnLeft();
9  moveForward();
10 if (isClearRight()) {
11   turnRight();
12   moveForward();
13   moveForward();
14 }
15 if (isClearRight()) {
16   turnRight();
17   moveForward();
18   moveForward();
19 }
20 turnRight();
21 moveForward();
22 if (isClearLeft()) {
23   turnLeft();
24   moveForward();
25   moveForward();
26   moveForward();
27   moveForward();
28 }
29 if (isClearRight()) {
30   turnRight();
31   moveForward();
32   moveForward();
33 }
34 turnRight();
35 moveForward();
```

11. Conditionals #11

Solution:

```
1  if (isClearRight()) {
2    turnRight();
3    moveForward();
4  }
5  turnLeft();
```

```

6 moveForward();
7 turnRight();
8 moveForward();
9 if (isClearLeft()) {
10     turnLeft();
11     moveForward();
12 moveForward();
13 moveForward();
14 }
15 if (isClearLeft()) {
16     turnLeft();
17     moveForward();
18 moveForward();
19 }
20 if (isClearRight()) {
21     turnRight();
22     moveForward();
23 moveForward();
24 }
25 pullLever();
26 turnAround();
27 if (isClearAhead()) {
28     moveForward();
29 moveForward();
30 }
31 if (isClearLeft()) {
32     turnLeft();
33     moveForward();
34 moveForward();
35 }
36 if (isClearLeft()) {
37     turnLeft();
38     moveForward();
39 moveForward();
40 }
41 if (isClearRight()) {
42     turnRight();
43     moveForward();
44 moveForward();
45 }

```

12. Conditionals #12

Solution:

```

1 if (isClearRight()) {
2     turnRight();
3     moveForward();
4 }
5 turnRight();
6 moveForward();
7 turnLeft();
8 moveForward();
9 moveForward();
10 turnLeft();
11 moveForward();
12 turnRight();

```

```
13 moveForward();
14 moveForward();
15 turnRight();
16 moveForward();
17 moveForward();
18 pullLever();
19 turnAround();
20 moveForward();
21 moveForward();
22 if (isClearLeft()) {
23   turnLeft();
24   moveForward();
25   moveForward();
26 }
27 turnLeft();
28 moveForward();
29 if (isClearRight()) {
30   turnRight();
31   moveForward();
32   moveForward();
33   moveForward();
34 }
35 turnLeft();
36 moveForward();
37 moveForward();
38 turnLeft();
39 moveForward();
40 turnRight();
41 moveForward();
42 turnLeft();
43 moveForward();
44 pullLever();
45 turnLeft();
46 moveForward();
47 turnRight();
48 moveForward();
49 moveForward();
50 turnRight();
51 moveForward();
52 turnLeft();
53 moveForward();
54 pullLever();
55 turnAround();
56 moveForward();
57 turnLeft();
58 moveForward();
59 turnRight();
60 moveForward();
```

13. Conditionals #13

Solution:

```
1 if (isClearRight()) {
2   turnRight();
```

```

3   moveForward();
4 }
5 if (isClearRight()) {
6   turnRight();
7   moveForward();
8   moveForward();
9 }
10 turnRight();
11 moveForward();
12 pullLever();
13 turnAround();
14 moveForward();
15 moveForward();
16 turnRight();
17 moveForward();
18 if (isClearLeft()) {
19   turnLeft();
20   moveForward();
21   moveForward();
22 }
23 if (isClearLeft()) {
24   turnLeft();
25   moveForward();
26   moveForward();
27 }
28 turnRight();
29 moveForward();
30 turnLeft();
31 moveForward();
32 pullLever();
33 turnAround();
34 moveForward();
35 turnLeft();
36 moveForward();
37 if (isClearRight()) {
38   turnRight();
39   moveForward();
40 }

```

14. Conditionals #14

Solution:

```

1 if (isClearRight()) {
2   turnRight();
3   moveForward();
4 }
5 if (isClearRight()) {
6   turnRight();
7   moveForward();
8   moveForward();
9   moveForward();
10 }
11 if (isClearLeft()) {
12   turnLeft();
13   moveForward();
14   moveForward();
15 }

```

```
16 pullLever();
17 if (isClearLeft()) {
18     turnLeft();
19     moveForward();
20 moveForward();
21 }
22 if (isClearLeft()) {
23     turnLeft();
24     moveForward();
25 moveForward();
26 }
27 turnRight();
28 moveForward();
29 if (isClearLeft()) {
30     turnLeft();
31     moveForward();
32 moveForward();
33 }
34 if (isClearRight()) {
35     turnRight();
36     moveForward();
37 moveForward();
38 }
39 turnRight();
40 moveForward();
41 turnLeft();
42 moveForward();
43 if (isClearRight()) {
44     turnRight();
45     moveForward();
46 moveForward();
47 moveForward();
48 }
```