

# Intro to Conditionals



By the end  
of this  
lesson, you  
should be  
able to...



By the end  
of this  
lesson, you  
should be  
able to...



By the end  
of this  
lesson, you  
should be  
able to...

1. Use the "if", "else", and "else if" keywords



By the end  
of this  
lesson, you  
should be  
able to...

1. Use the "if", "else", and "else if" keywords
2. Use conditionals inside functions



By the end  
of this  
lesson, you  
should be  
able to...

1. Use the "if", "else", and "else if" keywords
2. Use conditionals inside functions
3. Combine comparison operators and expressions with conditionals!



# Human conditionals (pseudocoding)



# Human conditionals (pseudocoding)



# Human conditionals (pseudocoding)

If I am tired then I want some coffee



# Human conditionals (pseudocoding)

If I am tired then I want some coffee

Does tired = true?



# Human conditionals (pseudocoding)

If I am tired then I want some coffee

Does tired = true?

Then buy some coffee



# Human conditionals (pseudocoding)

If I am tired then I want some coffee

Does tired = true?

Then buy some coffee

Does tired = false?



# Human conditionals (pseudocoding)

If I am tired then I want some coffee

Does tired = true?

Then buy some coffee

Does tired = false?

Then don't buy some coffee



# Human conditionals (pseudocoding)

If I am tired then I want some coffee

Does tired = true?

Then buy some coffee

Does tired = false?

Then don't buy some coffee

If I am broke, then I can't buy coffee



# Human conditionals (pseudocoding)

If I am tired then I want some coffee

Does tired = true?

Then buy some coffee

Does tired = false?

Then don't buy some coffee

If I am broke, then I can't buy coffee

Does broke = true?



# Human conditionals (pseudocoding)

If I am tired then I want some coffee

Does tired = true?

Then buy some coffee

Does tired = false?

Then don't buy some coffee

If I am broke, then I can't buy coffee

Does broke = true?

Then no coffee



# Human conditionals (pseudocoding)

If I am tired then I want some coffee

Does tired = true?

Then buy some coffee

Does tired = false?

Then don't buy some coffee

If I am broke, then I can't buy coffee

Does broke = true?

Then no coffee

Does broke = false?



# Human conditionals (pseudocoding)

If I am tired then I want some coffee

Does tired = true?

Then buy some coffee

Does tired = false?

Then don't buy some coffee

If I am broke, then I can't buy coffee

Does broke = true?

Then no coffee

Does broke = false?

Then coffee!!



# if statement

```
if(condition) {  
    do if true  
}
```



# if statement

```
if(condition) {  
    do if true  
}
```

```
if(3 > 0) {  
    console.log("this is true");  
}
```

## IMPORTANT

*The expression inside of the parentheses will be evaluated as a boolean.*



# if statement

```
if(condition) {  
    do if true  
}
```

```
if(3 > 0) {  
    console.log("this is true");  
}
```

## IMPORTANT

*The expression inside of the parentheses will be evaluated as a boolean.*



What punctuation marks look familiar?



# if statement

```
if(condition) {  
    do if true  
}
```

```
if(3 > 0) {  
    console.log("this is true");  
}
```

## IMPORTANT

*The expression inside of the parentheses will be evaluated as a boolean.*



What punctuation marks look familiar?



What do you notice about the indentation?



# What do you think happens if the if condition is not met?

// syntax:

```
if(condition) {  
    //do something  
}
```

// example:

```
if(3 < 0) {  
    console.log("do this if true");  
}
```



# What do you think happens if the if condition is not met?

// syntax:

```
if(condition) {  
    //do something  
}
```

// example:

```
if(3 < 0) {  
    console.log("do this if true");  
}
```



# What do you think happens if the if condition is not met?

// syntax:

```
if(condition) {  
    //do something  
}
```

// example:

```
if(3 < 0) {  
    console.log("do this if true");  
}
```

Then nothing happens!



# What about something else?

// syntax:

```
if(condition) {  
    //do something  
} else {  
    //do something else  
}
```



# What about something else?

// syntax:

```
if(condition) {  
    //do something  
} else {  
    //do something else  
}
```

// example:

```
if(3 > 0) {  
    console.log("do this if true");  
} else {  
    console.log("do this instead");  
}
```



# What if there are multiple conditions?

```
if(someCondition1) {  
    // do something  
} else if (someCondition2) {  
    // do something else  
} else {  
    // do if no conditions are met  
}
```



# Conditionals inside functions

```
function bigNumber(numA, numB) {  
  if (numA > numB) {  
    return "A is big!";  
  } else {  
    return "B is big!";  
  }  
}  
  
console.log(bigNumber(4, 89));
```

Take one minute to figure out what is happening.

What do you think will get printed?



# Things Developers Say:

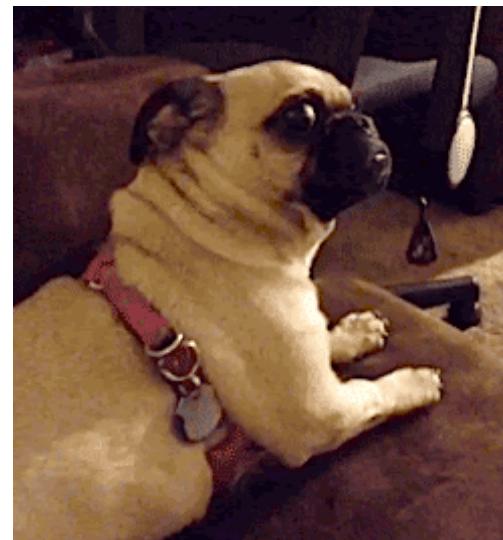
Make your code  
readable.

Why would they say this?



# Indentation matters!

```
function bigNumber(numA, numB) { if (numA > numB) { return  
"A is big!"; } else { return "B is big!"; } }  
  
console.log(bigNumber(4, 89));
```



The body (anything between the curly brackets) should be indented clearly and properly.

```
function bigNumber(numA, numB) {  
    if (numA > numB) {  
        return "A is big!";  
    } else {  
        return "B is big!";  
    }  
  
    console.log(bigNumber(4, 89));
```



```
function whichIsBig(A, B, C, D, E, F, G) { if (A > B)
{ return "A is big!"; } else if (B > C) { return "B is
big!"; } else if (C > D) { return "C is big!"; } else
if (D > E) { return "D is big!"; } else if (E > F) {
return "E is big!"; } else if (F > G) { return "F is
big!"; } else { return "G is big!"; } }
```

We write code for  
computers...



```
function whichIsBig(A, B, C, D, E, F, G) {  
    if (A > B) {  
        return "A is big!";  
    } else if (B > C) {  
        return "B is big!";  
    } else if (C > D) {  
        return "C is big!";  
    } else if (D > E) {  
        return "D is big!";  
    } else if (E > F) {  
        return "E is big!";  
    } else if (F > G) {  
        return "F is big!";  
    } else {  
        return "G is big!";  
    }  
}
```

AND  
for humans!



# REVIEW TIME



# REVIEW TIME



# REVIEW TIME

1. Conditions must always evaluate to what kind of data type?



# REVIEW TIME

1. Conditions must always evaluate to what kind of data type?
2. What punctuation is used around a condition?



# REVIEW TIME

1. Conditions must always evaluate to what kind of data type?
2. What punctuation is used around a condition?
3. If a condition is not met, what keyword do you use in a conditional?



# REVIEW TIME

1. Conditions must always evaluate to what kind of data type?
2. What punctuation is used around a condition?
3. If a condition is not met, what keyword do you use in a conditional?
4. How do you specify multiple conditions?



# REVIEW TIME

1. Conditions must always evaluate to what kind of data type?
2. What punctuation is used around a condition?
3. If a condition is not met, what keyword do you use in a conditional?
4. How do you specify multiple conditions?
5. Why does indentation matter?



# Think / Pair / Share

1. Grab a small whiteboard. You have 2 minutes to answer the questions to the right.
2. Then, you have 2 minutes to share your answers with your partner. Agree on an answer.
3. After that, we will take a few minutes to talk about your answers with the whole class.

What happens when we invoke this function with a 9 ?

```
function whichCondition(number) {  
  if (number / number == false) {  
    return "A";  
  } else if (number - 1 > 7) {  
    return "B";  
  } else {  
    return "C";  
  }  
}  
  
console.log(whichCondition(9));
```



# Activity

Get started with *Intro to Conditionals!*

