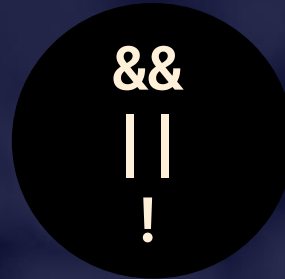
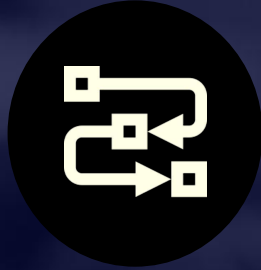
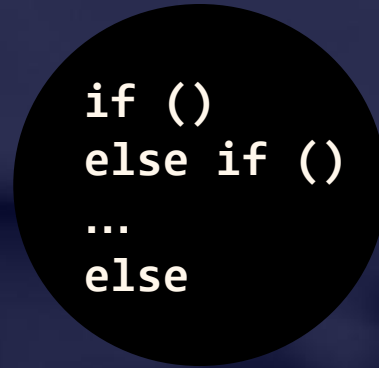
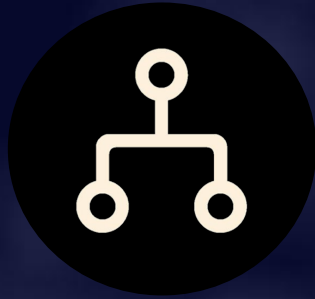




KINGSLAND
UNIVERSITY

Conditional Statements Advanced



Nested & Complex Conditional Statements



Table of Contents

- Review and Introduction
- Nested **if-else** Statements
 - **if-else** inside another **if-else**
- Logical Operators **&&**, **||**, **!** and **()**
- Conditional Statement **switch-case**
 - Using Multiple Labels in **switch-case**
- Practical Coding Exercises





Review

Conditional Statements



Comparison Operators

- Comparison operators work for **numbers**, **strings** and other **comparable data types**
 - Equal value (==), and equal type (===)
 - Not equal value(!=), and not equal type (!==)
 - Greater than (>), greater than or equal to (>=)
 - Less than (<), less than or equal to (<=)

```
console.log('5' == 5);    // true  
console.log(5 <= 3);      // false
```



Conditional Statements

- The **if-else** statement can be in a series

```
if (...)  
    // Some code  
else if (...)  
    // Some code  
else  
    // Some code
```

If one condition is true, the program will NOT check the rest of the conditions



Advanced Conditional Statements

Introduction



Real Life Example: Marketplace

- A market offers different **prices** for certain **products** based on the **day of the week**:

Product	Weekday	Weekend
Banana	2.50	2.70
Apple	1.30	1.60
Kiwi	2.20	3.00

- Calculate the price for certain product and day



Pseudocode: Marketplace

Read the input

If product is banana

 If it is a weekday the price is 2.50

 Otherwise the price is 2.70

If product is apple

 If its a weekday the price is 1.30

 Otherwise the price is 1.60

If product is kiwi

 If its a weekday the price is 2.20

 Otherwise the price is 3.00

```
if ()  
else if ()  
...  
else
```

Nested Conditions

If-Else inside another If-Else



Nested Conditional Statements

- An **if-else** statement can be **nested** within another **if-else** statement

```
if (expression) {  
    if (nested expression)  
        // Some code  
    else  
        // Some code  
}
```



Nested Conditional Statements (2)

- Only if the first condition is **true** the nested one is checked

```
if (expression) {  
    if (nested expression)  
        // Some code  
    else  
        // Some code  
}
```

Executes when the nested expression is false

- Deep nesting is not recommended
 - Use up to **3 nested levels**



Problem: Marketplace

- Write a function which receives a **product** and **day**
- Print the **price**, formatted to 2nd digit, based on the price table:

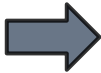
Product	Weekday	Weekend
Banana	2.50	2.70
Apple	1.30	1.60
Kiwi	2.20	3.00

Kiwi
Weekday



2.20

Banana
Weekend



2.70



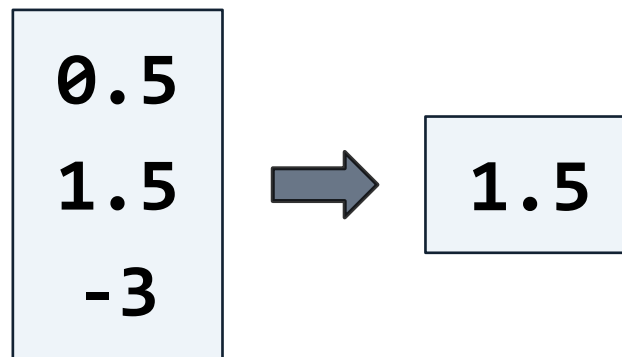
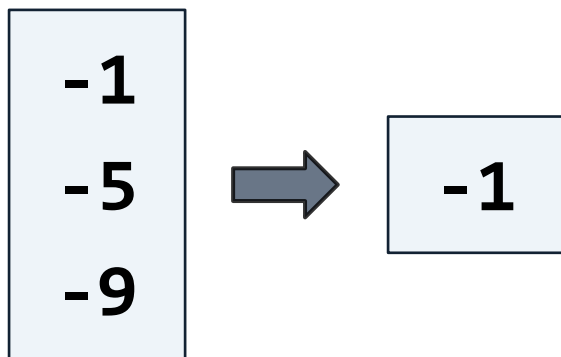
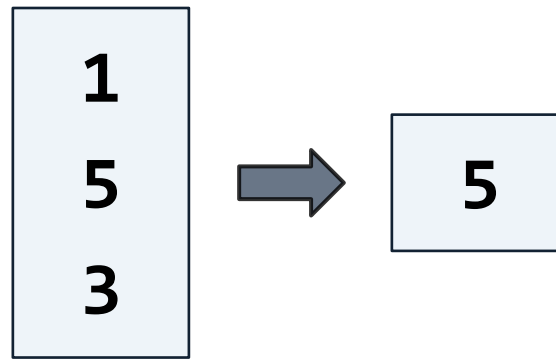
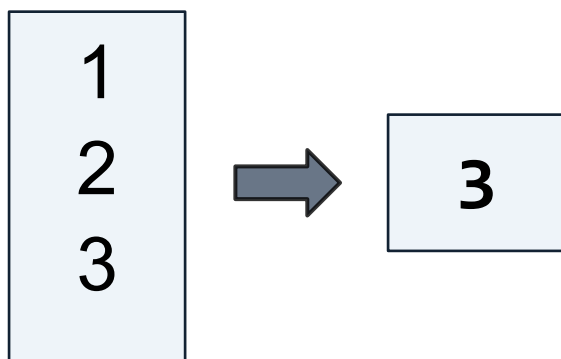
Solution: Marketplace

```
function marketplace(product, dayOfWeek) {  
  if (product == "Banana")  
    if (dayOfWeek == "Weekday")  
      console.log("2.50");  
  else  
    console.log("2.70");  
  else if (product == "Apple")  
    if (dayOfWeek == "Weekday")  
      console.log("1.30");  
  else  
    console.log("1.60");  
  // TODO: the same logic for "kiwi"  
}
```

```
marketplace(  
  "kiwi", "Weekday"  
);
```

Problem: Biggest of Three Numbers

- Write a function which takes **3 numbers** and prints **the biggest** number of them



Solution: Biggest of Three Numbers

```
function biggestNumberOfThree(n1, n2, n3) {  
  if (n1 > n2)  
    if (n1 > n3)  
      console.log(n1);  
    else  
      console.log(n3);  
  else  
    if (n2 > n3)  
      console.log(n2);  
    else  
      console.log(n3);  
}
```

$n1 > n2$
 $n1 > n3$

$n3 \geq n1 > n2$

$n2 \geq n1$
 $n2 > n3$

$n3 \geq n2 \geq n1$

NOT (!)
AND (&&)
OR (||)

Logical Operators

Checking Complex Conditions

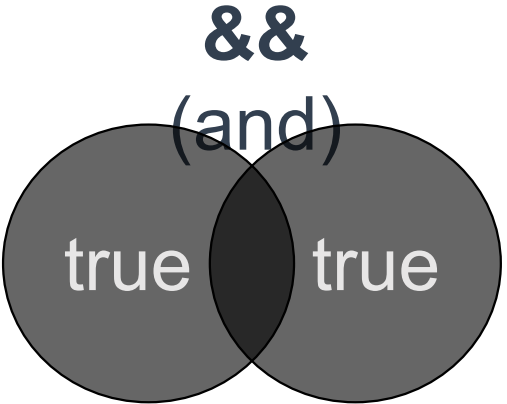


Logical Operators

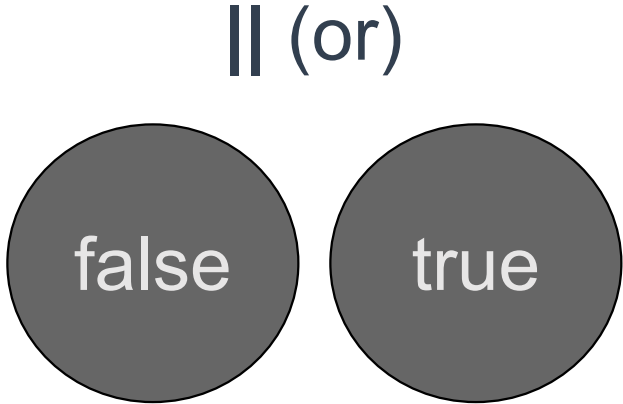
- Used to perform logical operations
- The logical operators in JavaScript are:
 - Logical **AND** (&&)
 - Logical **OR** (||)
 - Logical **NOT** (!)
- Brackets () change the order



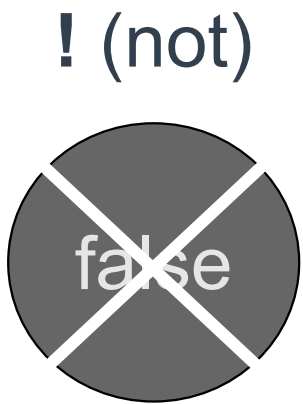
Logical Operators: Explanation



**Both
conditions
must be true
(intersection)**



**One
condition
must be true
(union)**



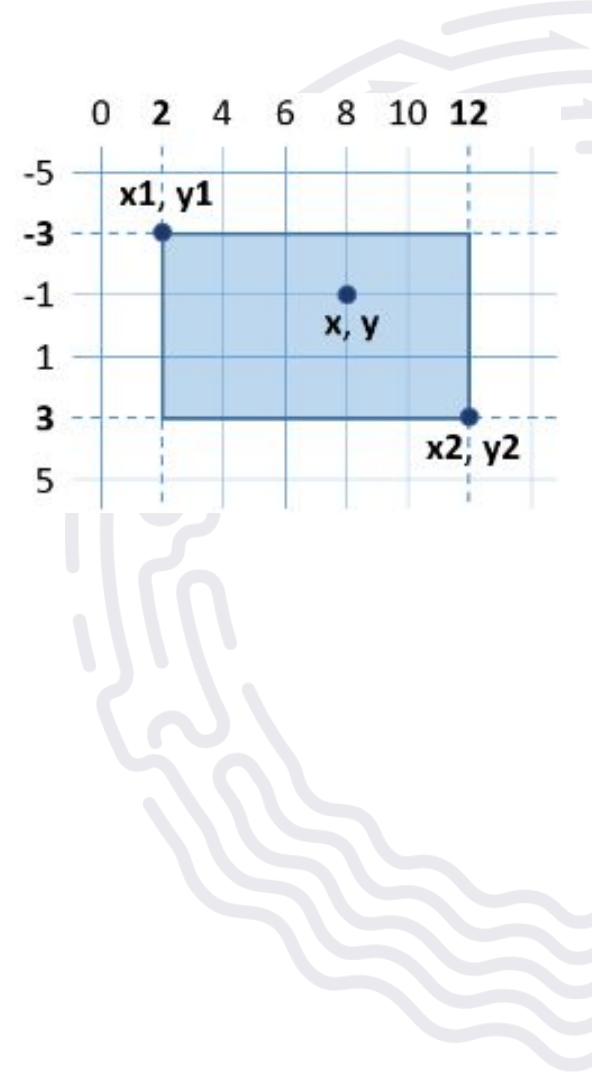
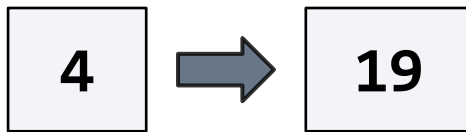
**Logical
negation
(inverse)**

Logical AND (&&)

- Returns the Boolean value **true** if all of the operands are **true** and **false** otherwise

```
if (x >= x1 && x <= x2 && y >= y1 && y <= y2)
```

- Write a function to **add bonus** to given points
 - If points are between 0 and 3, adds 5
 - If points are between 4 and 6, adds 15
 - If points are between 7 and 9, adds 20





Solution: Bonus Points

```
function bonusPoints(points) {  
  if (points >= 0 && points <= 3)  
    points += 5;  
  else if (points >= 4 && points <= 6)  
    points += 15;  
  else if (points >= 7 && points <= 9)  
    points += 20;  
  console.log(points);  
}  
bonusPoints(4); // 19
```





Logical OR (||)

- The result of the expression is **true** if one of the operands is **true**, otherwise the result is **false**

```
s === "tea" || s === "water" || s === "spaghetti"
```

- Problem: **check for food or drink**
 - Receive single parameter and print "**drink**", "**food**" or "**unknown**"
 - **Foods**: curry, noodles, sushi, spaghetti
 - **Drinks**: tea, water, coffee
 - Everything else is unknown



Solution: Food or Drink

```
function foodOrDrink(s) {  
  if (s === "curry" || s === "noodles" ||  
      s === "sushi" || s === "spaghetti")  
    console.log("food");  
  else if (s === "tea" || s === "water" ||  
           s === "coffee")  
    console.log("drink");  
  else  
    console.log("unknown");  
}  
foodOrDrink("sushi"); // food
```





Logical NOT (!)

- Logical negation returns **true** when the operand is **false**, and **false** when the operand is **true**
- Example: **check for valid number**
 - A number is **valid** if is in the range [100...200] or is equal to 0

```
let valid =  
    (num >= 100 && num <= 200) || num == 0;  
if (!valid)  
    console.log("invalid");
```




Switch-Case

Checking Multiple Values for the Same Input



The Switch-Case Statement

- Used for choosing among a list of possibilities
- Alternative to an **if-else** series of statements

```
switch (selector) {  
    case value1:  
        statements;  
        break;  
    default:  
        statements;  
        break;  
}
```



Switch-Case Example: Print Yes / No

```
function yesOrNo(choice) {  
  switch (choice) {  
    case "Y":  
      console.log("Yes");  
      break;  
    case "N":  
      console.log("No");  
      break;  
    default:  
      console.log("Invalid response");  
      break;  
  }  
}  
yesOrNo("Y"); // Yes
```



Multiple Labels

Same Action for Several Values



Multiple Labels in Switch-Case

- Same logic may apply for more than one case

```
switch (selector) {  
    case value1:  
    case value2:  
        statements;  
        break;  
    default:  
        statements;  
        break;  
}
```



Multiple Labels: Example

```
function animals(animal) {  
  switch (animal) {  
    case "dog":  
    case "cat":  
      console.log("mammal");  
      break;  
    default:  
      console.log("unknown");  
      break;  
  }  
}  
animals("dog"); // mammal
```





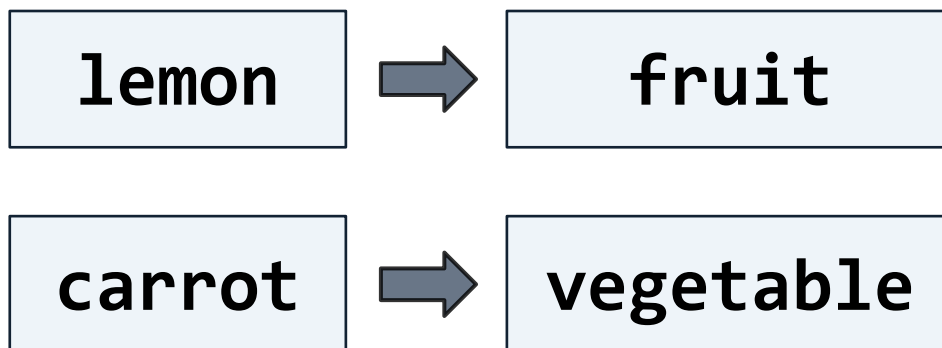
Problem Solving

Practical Coding Exercises



Problem: Fruit or Vegetable

- Write a function to **check for fruit or vegetable**:
 - Receive an **item** from the greengrocery
 - **Fruits**: banana, apple, kiwi, cherry, lemon, grapes
 - **Vegetables**: cucumber, pepper, carrot, onion
 - Print: "**vegetable**", "**fruit**" or "**unknown**"





Solution: Fruit or Vegetable

```
function fruitOrVegetable(product) {  
  switch (product) {  
    case "cucumber":  
    case "pepper":  
    case "carrot":  
      console.log("vegetable");  
      break;  
    //TODO: Implement the other cases  
  }  
}
```

```
fruitOrVegetable("cucumber");
```



Problem: Day of Week

- Write a function to **print the day of week as words**:
 - Receives integer **n**: the **day of the week** in range [1..7]
 - Prints the **name of the day** (as words, in English)
 - Prints "**Error**" if the number is not in the given range

1	➔	Monday	0	➔	Error
7	➔	Sunday	5	➔	Friday
9	➔	Error	2	➔	Tuesday



Solution: Day of Week

```
function dayOfWeek(day) {  
  switch (day) {  
    case 1: console.log("Monday"); break;  
    case 2: console.log("Tuesday"); break;  
    //TODO: Implement the other valid days  
    default: console.log("Error"); break;  
  }  
}
```

```
dayOfWeek(7); // Sunday  
dayOfWeek(9); // Error
```

Problem: Vowel or Consonant

- Write a function to **check a letter for vowel or consonant**:
 - Receives a letter from the English alphabet
 - Print either "Vowel" or "Consonant"

a → Vowel

x → Consonant

E → Vowel

Z → Consonant

i → Vowel

b → Consonant



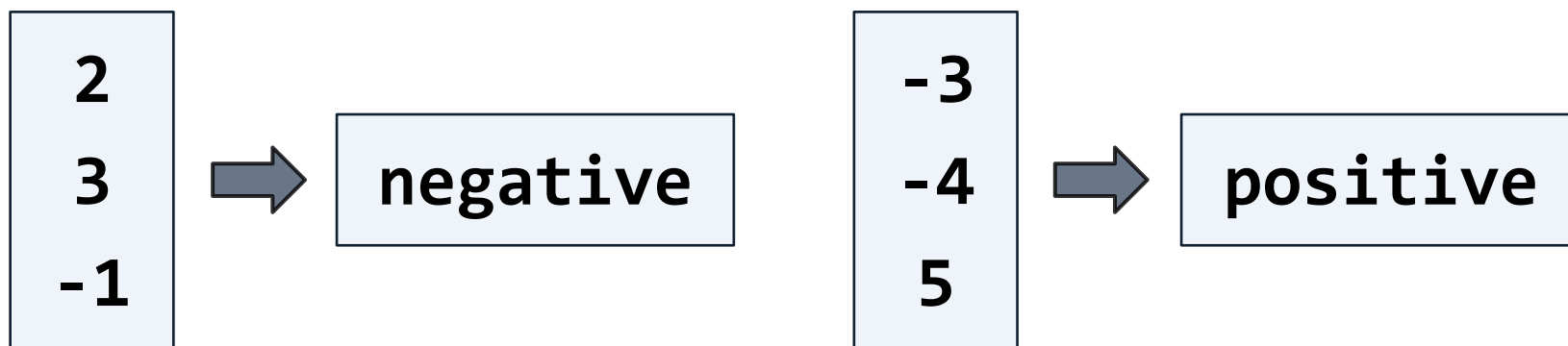
Solution: Vowel or Consonant

```
function vowelOrConsonant(letter) {  
  if (letter === 'A' || letter === 'a' ||  
      letter === 'E' || letter === 'e' ||  
      letter === 'I' || letter === 'i' ||  
      letter === 'O' || letter === 'o' ||  
      letter === 'U' || letter === 'u')  
    console.log("Vowel");  
  else  
    console.log("Consonant");  
}
```

```
vowelOrConsonant('A');
```

Problem: Product of 3 Numbers' Sign

- Calculate the **sign of the product of 3 numbers**:
 - Function should receive 3 floating-point numbers
 - Print the **sign of the product** of the entered 3 numbers: **positive**, **negative** or **zero**
- Try to do this without multiplying the 3 numbers





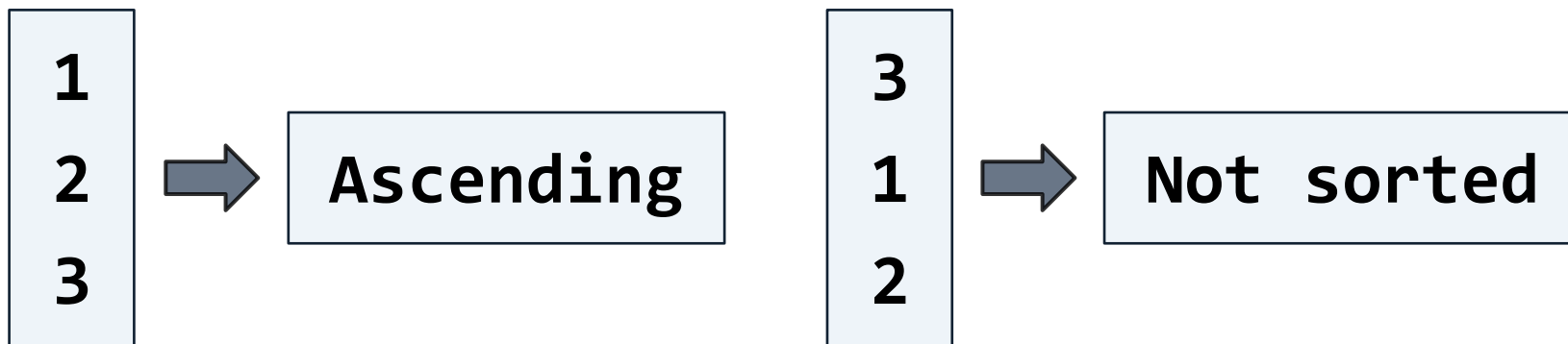
Solution: Product of 3 Numbers' Sign

```
function productSign(n1, n2, n3) {  
  if (n1 === 0 || n2 === 0 || n3 === 0)  
    console.log("zero");  
  else {  
    let negativeNumbersCount = 0;  
    if (n1 < 0) negativeNumbersCount++;  
    if (n2 < 0) negativeNumbersCount++;  
    if (n3 < 0) negativeNumbersCount++;  
    if (negativeNumbersCount % 2 === 0)  
      console.log("positive");  
    else  
      console.log("negative");  
  }  
}
```

```
product(1, 2, 3);
```

Problem: Sorted Numbers

- Write a function, which checks for **sorted 3 numbers**:
 - Receives 3 real numbers
 - Prints "**Ascending**" if the numbers are in ascending order
 - Prints "**Descending**" if the numbers are in descending order
 - Prints "**Not sorted**" in any other case





Solution: Sorted Numbers

```
function sortedNumbers(n1, n2, n3) {  
  if (n1 < n2 && n2 < n3)  
    console.log("Ascending");  
  else if (n1 > n2 && n2 > n3)  
    console.log("Descending");  
  else  
    console.log("Not sorted");  
}
```

```
sortedNumbers(1, 2, 3);  
sortedNumbers(2, 1, 3);
```

Problem: Vacation Expenses

- Write a function, which **calculates vacation expenses**:
 - Receives **season**, **accommodation type** and count of the **days**
 - Prints the **total expenses**, based on the **price table** below, formatted to the 2nd digit after the decimal point

Season	Hotel	Camping	Discount
Spring	30	10	20%
Summer	50	30	0%
Autumn	20	15	30%
Winter	40	10	10%

Winter
Hotel
5



180.00



Solution: Vacation Expenses

```
function vacationExpenses(season, accommodation, days) {  
  if (season === "Spring") {  
    if (accommodation === "Hotel")  
      totalPrice = days * 30 * 0.80;  
    else if (accommodation === "Camping")  
      totalPrice = days * 10 * 0.80;  
  }  
  // TODO: Implement the other cases and print the sum  
}
```

```
vacationExpenses("Winter", "Hotel", 5);
```

Problem: Cinema

- Calculate the price for all the tickets for a cinema movie:
 - A function receives the **type** of the movie (string), the **rows** (number) and the **seats per row** (number) in the cinema
 - Prints the **total price** for all seats formatted to the 2nd digit after the decimal point

Type	Price
Premiere	12.00
Normal	7.50
Discount	5.00

Price for
one seat

Normal
12
9

$7.50 * 12 * 9$

810.00



Summary

- An if-else statement can be nested within another if-else
- Logical operators operate over boolean expressions
 - && (and), || (or), ! (not)
- The switch-case statement is an alternative to the if-else





Questions?





License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © Kingsland University – <https://kingslanduniversity.com>





THANK YOU