iOS Course

Second class - conditionals, collections, loops

Conditionals

Conditionals

In computer science, conditional statements are features of a programming language, which perform different computations or actions depending on whether a programmer-specified boolean condition evaluates to true or false.

- if/else
- else if
- switch

Table 3.1 Comparison operators

Operator	Description
<	Evaluates whether the value on the left is less than the value on the right.
<=	Evaluates whether the value on the left is less than or equal to the value on the right.
>	Evaluates whether the value on the left is greater than the value on the right.
>=	Evaluates whether the value on the left is greater than or equal to the value on the right.
==	Evaluates whether the value on the left is equal to the value on the right.
!=	Evaluates whether the value on the left is not equal to the value on the right.
===	Evaluates whether the two references point to the same instance.
!==	Evaluates whether the two references do not point to the same instance.

Table 3.2 Logical operators

Operator	Description
&&	Logical AND: true if and only if both are true (false otherwise).
||	Logical OR: true if either is true (false only if both are false).
!	Logical NOT: evaluates whether a condition is false (returns true for a false operand and vice versa).

Conditional statements

Conditional statements

In computer science, conditional statements are features of a programming language, which perform different computations or actions depending on whether a programmer-specified boolean condition evaluates to true or false.

- if/else
- else if
- switch

Collections

Collections

Swift provides three primary collection types, known as arrays, sets, and dictionaries, for storing collections of values.

- ***Arrays-*** are ordered collections of values.
- ***Sets-*** are unordered collections of unique values.
- ***Dictionaries-*** are unordered collections of key-value associations.

Collection Type	Ordered?	Unique?	Stores
Array	Yes	No	Elements
Dictionary	No	Keys	Key-value pairs
Set	No	Elements	Elements

Loops

Loops

Loops help with repetitive tasks. They execute a set of code repeatedly, either for a given number of iterations or for as long as a defined condition is met.

- ***For***
- ***While***

Challenges