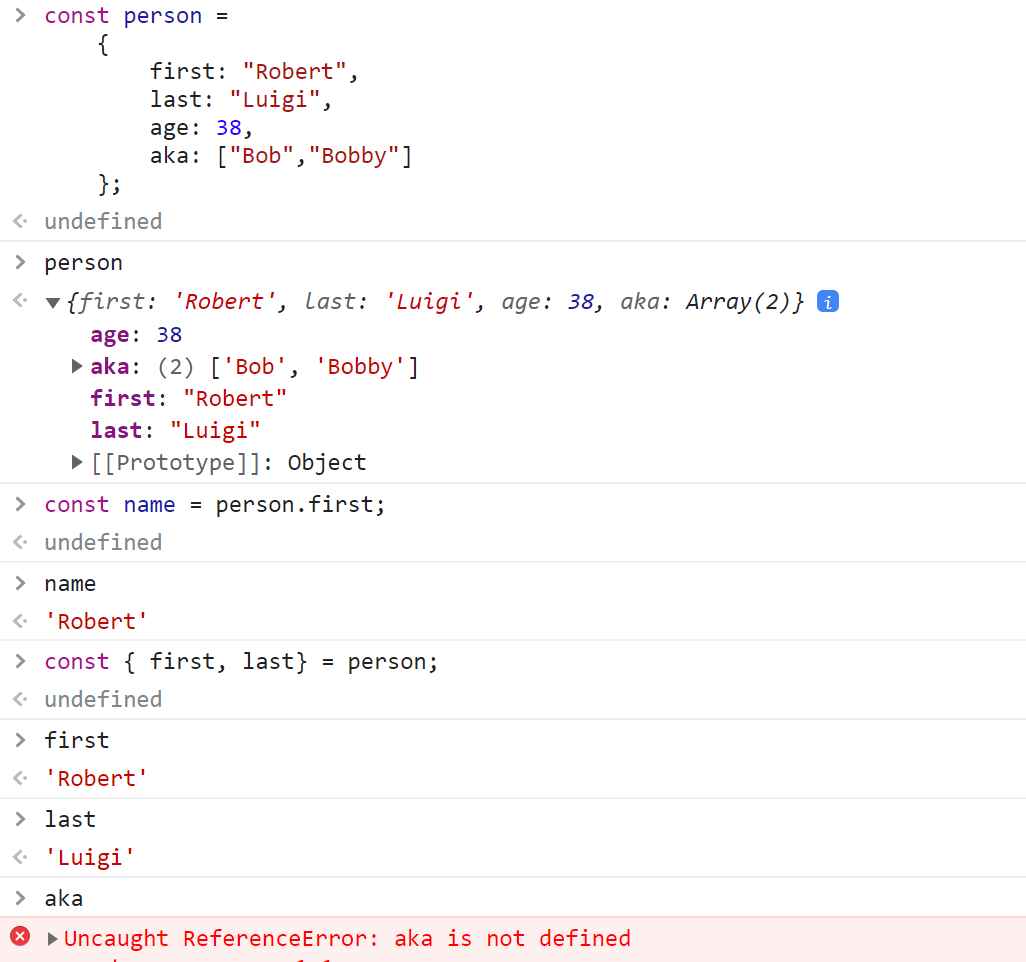
Monday Lecture

Invented in 1995, originally known as LiveScript. Known as ECMAScript.

* ECMAScript is abbreviated as ES
* Early versions of ECMAScript used the ES abbreviation
* ES6 or ECMA2015
* JavaScript destructuring allows extracting data from object data types (arrays, objects, and newer EDS6 types maps and sets) into their own variables using a simple syntax
  + an array is an object
  + objects have properties and methods
* creating object, example with property variables
* object destructing: data type { list of variables} = object
* variables can be renamed
* or given default values if not defined
* default values do not apply to null, false or 0



Graphical user interface, text, application, email

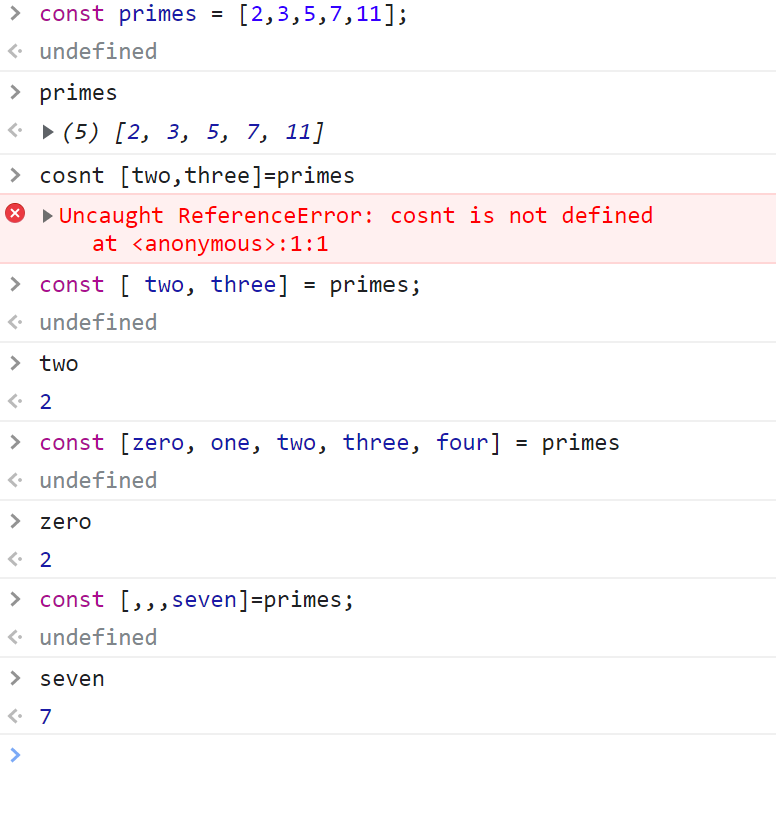
Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Array destructuing uses the array square bracket syntax and array position (index) to assign variables

* array items can be skipped



Modularization: reduced code into smaller more manageable components

Complexity reduction: smaller components means reduced individualized component complexity

Encapsulation: isolates functionality and programming required to support that functionality, including variables to include components

Debugging smaller components are easirer to debug and easier to code.

Let and const

* block level scope
* global: defined outside of any block level scope
* scope: defined within a block using curly braces
  + Functions
  + Conditions
  + Loops
  + Arbitrary/ad hoc blocks

Node moduelses

* Module file creates a module.exports object
* A module code is encapsulated within a single file, and exported for use by another file

Importing modules:

* Only import the items needed from the code module

Wednesday Lecture

Normal function

Function foo (parameter) { }  
Anonymous functions, used as a callback

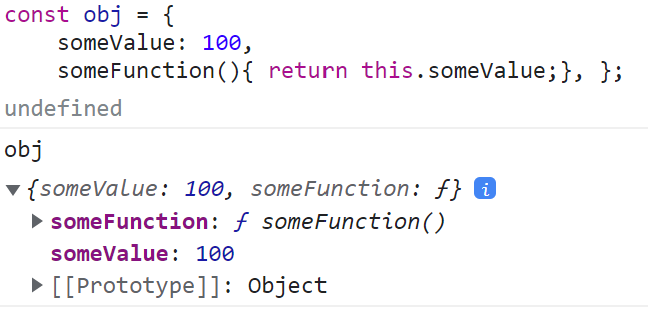
Function (parameter) { }

Function expressions, using anonymous functions

Const foo = function (parameter) { }

Parameters => function body

Function body can implicit return or explicit return



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

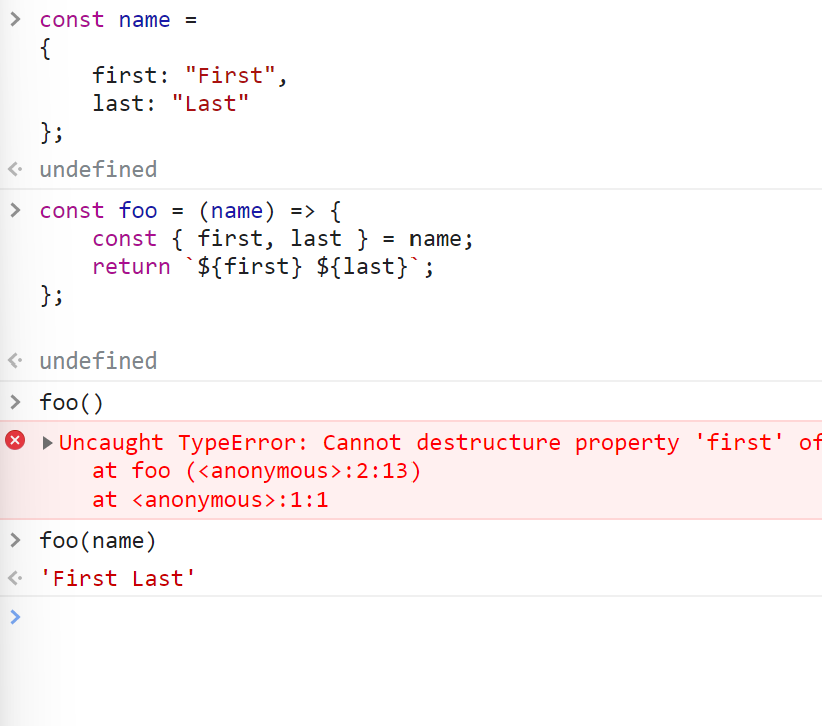
Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Explicit Return Arrow Functions



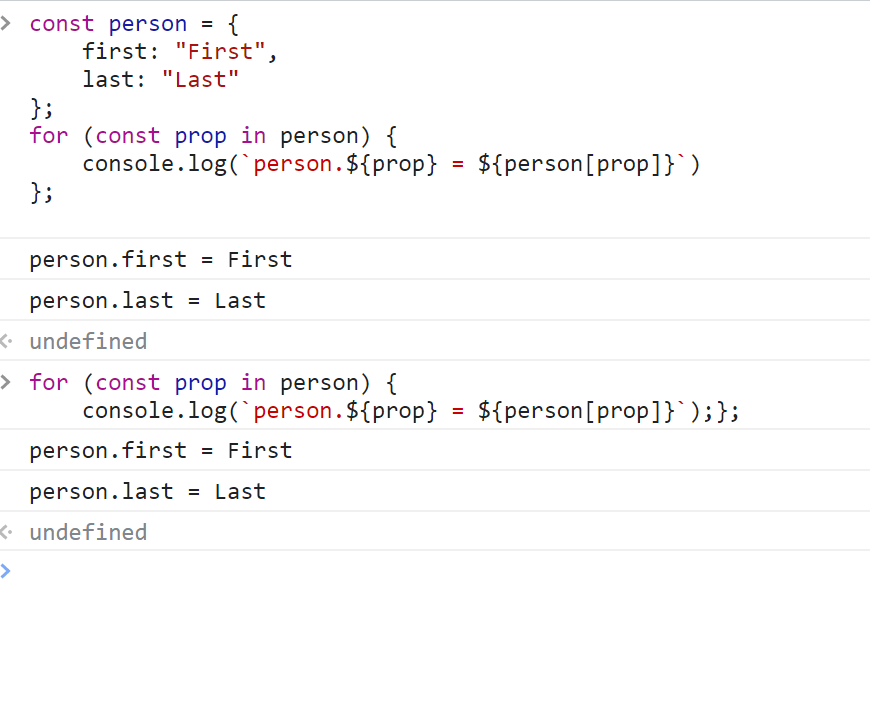
Arrow Functions as Methods

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



For array loop

Graphical user interface, text, application, email

Description automatically generated

Default parameters:

-defaults simplified code,

-parameters may have default values

Ellipsis:  
default destructure values:

The destructuring syntax also supports default values

Ellipsis:

Rest: collect or gather object elements

Spread: spread or scatter object elements

Graphical user interface, text, email

Description automatically generated

Red white blue example

for in loops