

Programming Paradigms

Imperative: Programming with an explicit sequence of commands that update state.

Declarative: Programming by specifying the result you want, not how to get it.

Structured: Programming with clean, goto-free, nested control structures.

Procedural: Imperative programming with procedure calls.

Functional (Applicative): Programming with function calls that avoid any global state.

Object-Oriented: Programming by defining objects that send messages to each other. Objects have their own internal (encapsulated) state and public interfaces. Object orientation can be: Class-based: Objects get state and behavior based on membership in a class. Prototype-based: Objects get behavior from a prototype object.

A programming paradigm is a style, or "way," of programming. Some languages make it easy to write in some paradigms but not others.

Higher-order functions

When we talk about higher-order functions, we mean a function that either:

- takes one or more functions as arguments, or
- returns a function as its result

They let us encapsulate variables, make HOCs in React, currying, etc.

Pure function

A pure function is a function which:

- Given the same input, will always return the same output.
- Produces no side effects.

Functions as first-class entities

Functions as first-class entities can:

- refer to it from constants and variables
- pass it as a parameter to other functions
- return it as result from other functions

Currying

Currying is a process to reduce functions of more than one argument to functions of one argument.

Serverless

- 1) Abstraction
- 2) Scaling
- 3) Effective costs

Redux

Redux is a state management tool. It offers a single store of state, which can be accessed from any React component.

There are three main principles of Redux:

- 1) **Single source of truth.** It means that the state of the app is stored as an object tree within a single store.
- 2) **State is read-only.** It means that the state can be changed only by emitting an action.

Redux (cont)

3) **Changes are made with pure functions.**

To specify how the state tree is transformed by actions, you write pure reducers.

Reducers are just pure functions that take the previous state and an action, and return the next state.

React

React is a JavaScript library for building user interfaces. It is the view layer for web applications. At the heart of all React applications are components.

Advantages:

1.1 Component Creation

React enables the creation of module-like pieces of code called "Components" which allows reusability.

1.2 Virtual DOM

Considered the next biggest leap in web development since AJAX, the virtual DOM (short for Document Object Model) is the core reason why React enables the creation of fast, scalable web apps.

Through React's memory reconciliation algorithm, the library constructs a representation of the page in a virtual memory, where it performs the necessary updates before rendering the final web-page into the browser.

1.3 Easy to learn

We need to clarify that React is NOT a framework; unlike Angular or Vue.js, but a library that is consistently used in association with other Javascript libraries. Hence, there is a shorter learning curve involved in understanding React compared to other comprehensive libraries.



By NikiSolo

cheatography.com/nikisolo/

Not published yet.

Last updated 8th July, 2019.

Page 1 of 2.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish

Yours!

<https://apollopadd.com>

Scrum

Scrum is a project management methodology.

Scrum relies on a self-organizing, cross-functional team. The Scrum model suggests that projects progress via a series of sprints. In keeping with an agile methodology, sprints are timeboxed to no more than a month long, most commonly two weeks. Scrum methodology advocates for a planning meeting at the start of the sprint, where team members figure out how many items they can commit to, and then create a sprint backlog – a list of the tasks to perform during the sprint.

Prototypal Inheritance

In JavaScript every object has a prototype object. These prototype objects are live objects that can be changed at any point in time. The most important thing to know about prototype objects is that they can be linked with each other to create chains. JavaScript internally can traverse these chains and look for methods and properties. So if you ask a random object about a property or method, it's going to traverse the chain to find them.

When a function is called with the new keyword, couple of things happen behind the scenes:

- 1) A new empty object is created
- 2) The context object this is bound to the new empty object
- 3) The new object is linked to the function's prototype property
- 4) this is automatically returned unless another value is returned explicitly from the function

this

In JavaScript, 'this' normally refers to the object which 'owns' the method, but it depends on how a function is called.

Global Scope

If there's no current object, 'this' refers to the global object. In a web browser, that's 'window' — the top-level object which represents the document, location, history and a few other useful properties and methods.

Agile

Agile software development refers to software development methodologies centered round the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. The ultimate value in Agile development is that it enables teams to deliver value faster, with greater quality and predictability, and greater aptitude to respond to change. Scrum and Kanban are two of the most widely used Agile methodologies. Below are the most frequently asked questions around Agile and Scrum, answered by our experts..



By NikiSolo

cheatography.com/nikisolo/

Not published yet.

Last updated 8th July, 2019.

Page 2 of 2.

Sponsored by **ApolloPad.com**

Everyone has a novel in them. Finish Yours!

<https://apollopad.com>