Initially, the declarative development will have very little interaction with the APIs.

Programmatic development will be where you will use them more often.

Salesforce APIs (application programming interfaces, duh) and what they do:

- SOAP - integrate your org's data with other apps, via SOAP protocols.
- REST - Access objects in your org, via REST protocols.
- Metadata - manage customization and build tools that manage your metadata model.
- Tooling - Build custom development tools for platform apps
- Marketing Cloud - Expose MC capabilities with REST api and get access to most email functionality with the SOAP api.
- Bulk - Load, delete and perform async queries on large data sets.
- Streaming - sending/receiving notifications. These can be for data changes and/or custom events.
- Chatter (REST)- build UI for Chatter, communities, recommendations, files, topics, and more.
- Mobile SDK - an SDK, not API, but integrate native or hybrid mobile apps with salesforce.

Heroku:

Built on AWS, very flexible from a language standpoint (java, python, php).
Heroku Connect - unifies Salesforce data with Heroku Postgres data so you don't have to manage moving info across platforms.

IoT, Bots, More:

These are examples of optional things you can incorporate into your development. Keep them in mind if they solve a business problem. Smart devices are on the rise.

API details and benefits:
Benefits:

- Outsourcing - allow you to outsource key data and functionality through a predictable interface. (ex: google maps. You focus on using the map in your app and don't worry about how the mapping works)
- Increased Mobility - receiving updates, migrating to new service, etc. the pattern of the interface may change, but you can easily adapt because of well-defined and documented standards. (*agreed upon standards are referred to as the API's **contract**)
- Abstraction - Similar to the outsourcing benefit, you only need to work with the resulting data/service that you are ingesting. If the provider makes changes on how they are handling things it does not affect you or your org function. That has been abstracted away.

- Increased Dev Productivity - allows developers to leverage code that is already developed and performing tasks. Minimizes the development time
- Use of HTTP Protocols to access Data (CRUD) - Post, Get, Put, Delete (verbs)