Hack

* Type checking
* Choose between dynamic and static typing
* Object Oriented
* Works with PHP scripts (designed to fix most problems with PHP)
* Library integration
* Created by Facebook team
* Open source
* Does it work?
  + Yes, but is a small niche
  + Improves PHP issues (works seamlessly with PHP scripts)
* Not supported much outside Facebook
* async {…}
  + Await expression
* How to use Hack language
  + Download as repository through Ubuntu
  + Create .hhconfig file
  + Use favorite editor and create a file with an extension of .hack
  + .hack file begins with <<\_EntryPoint>> at the top of the program
  + Method header with the following setup
    - Function (method name) (): (Return type)
  + Syntax of hack is like that of PHP and C#
  + Once completed, do hhvm (hack file name).hack to type check and compile the file

Dart

* Started creation in October 10, 2011
* Final Version November 14, 2013
* Made by Google Employees
  + Lars Bak- JavaScript Professional and for working on VMs
  + Kasper Lund
* Ecma International- standard organization for information and communication systems, made a technical committee
* Dart can be compiled into JavaScript
* Multiparadigm language (OO Scripting Language)
* Designed by Google to make programming language for apps to work on multiple platforms
* Now primarily used in the Flutter toolkit for making Android or iOS apps
* Variable Binding and Scope
  + Type safe language
  + Case sensitive
  + Supports:
    - Numbers
    - Strings
    - Booleans
    - Lists
    - Maps
* Is an optionally typed language
  + Ex. var number = 0; compiler knows this is an int
* Can trans-compile into JavaScript
* Very familiar and simple to pick up for Java developers
* Three types of constructors
  + Normal
  + Name
  + Factory

Ruby

* Created in 1993, release 1996 Yukihiro Matsumoto
* Improvement on the shortcomings of Perl and Python
* Features he wanted to implement
  + Simple syntax
  + Truly OO language
  + Iterators and closures
  + Exception handling
  + Garbage collection
* Became popular in 2005
* Used for
  + Hulu
  + Airbnb
  + Groupon
* \*\* (Exponentiation)
* :: (Constant Resolution Operator)
* Dynamic Typing
* Case Sensitive
* Left associative
* Pass-by-value
* Very similar syntax to Java and Python
* Semicolons are not required
* Interpreter already installed on Elvis
* Conventionally, Ruby files end in .rb
* Open source
* Lots of libraries and packages
* Good for web development
* Runtime speed is considered slow
* Not too popular anymore

Python

* Created in early 1990s, Guido Von Rossum, Netherlands
* Successor to ABC language, provide more modularity
* Has many contributors to language
* Used by
  + Intel
  + NASA
  + IBM
  + Pixar
  + Chase
* Use for
  + Web scripting
* Dynamically typed language
  + OO language
  + Scripting language but well worked
* Easy to learn- few keywords, simple structures
* Easy to read
* High level data types
* Databases
* Interpreter executes line by line
* Syntax details
  + No need to define a variable type
  + End of line terminates a statement
  + White space matters!
  + “def” is used to define a function
* White space principle also applies to functions
* Functions do not require a return type, return types are dynamically handled by interpreter
* High productive language with simple syntax
* Dynamically typed and flexibility
* Python does not compile, so it loses speed
* Requires more testing and is error prone, and errors only occur at runtime (since no compiler)
* Complicated unit testing
* Not native to mobile programming
* Achieves many of the goals it set out to do
* Very portable
* Useful for wide variety of applications
* Falls flat with safety and speed
* Does not do well with more in-depth niche problems

Crystal

* OO Language
* Statically typed (value of variable is determined at compile time)
* Syntax like Ruby
* Performance matches C/C++
* Relies on LLVM
* Designed by Ary Borenszweig, official release June 2014
* Latest version is 0.31.1 (not yet fully released)
* Purpose
  + Designed with Ruby syntax in mind
  + Speed, efficiency, and type safety of compiled language
  + Be able to call C code by writing bindings to it
* Shards- user made libraries
* Test space- built in testing library that can be called using “spec” command
* Formatting tool- allows you to format all program files in a directory ending in file extension .cr
* Documentation tool- reads all comments in every file in a directory ending in .cr, stores in “doc” directory
* Manual- extensive reference manual
* No need to declare what type the variables are
* No “for” loops, use “while” loops
* No compiler for Windows yet, must get Windows Subsystem from Microsoft Store

Perl

* General purpose high-level interpreted dynamically typed language
* Does not enforce programming paradigm
* Formed by “pulling together” best features of other languages such as sed, AWK, C, etc.
* Written by Larry Wall while working for NASA in 1987
* Designed to be easy for humans to write, rather than be easy for computers to understand
* Open source, believed to be free
* Why use Perl?
  + CGI Scripting- dynamically generated web pages
  + Original Task- extracting data from one source and translating it to another format
  + Designed mainly for text processing
  + Faster than Python
* Pragmas- like import statements
* Flexible, powerful, and ugly
* Good writability, bad readability

Swift

* Chris Lattner, 2010, with other Apply programmers
* First beta released in 2014
* Ideas were taken from Rust, Objective-C, Ruby, Haskell, C#, CLU
* Many versions, currently 5.1
* Same runtime as Objective-C
* Does not need header files
* Purpose
  + Easy to learn
    - No semicolons
    - Great first language for beginners
  + Designed for safety
    - Swift objects can never be nil, unless they are specified to be an optional
  + Fast and powerful
    - High-performance compiler
    - Fast for user
* Functional OO
* Statically typed
* Case sensitive
* Similar syntax to python, JavaScript
* Less error prone due to catching most of the errors at compile time rather than runtime
* = is assignment, == for comparison, === to check if two elements refer to the same object
* Automatic memory management and no pointers
* Uses Automatic Reference Counting (ARC) to manage memory, allows easier memory (de)allocation
* Pros
  + Faster to develop for and safer than Obj-C
  + Works with existing Obj-C
  + Cross-device support
* Cons
  + Still young, so changes can break older versions
  + XCode is the only true IDE for Swift, only on MacOS devices
  + No build for Windows that is supported by Apple
  + Cross-device only in Apple’s ecosystem

Go

* Aka Golang
* Developed in 2007
* Launched by Google, 2009
* Influenced by C/C++, Java, Python
* Features
  + Easy to learn
  + Fast compilation
  + Garbage Collection
  + Built-in Concurrency
  + Open source
  + Lots of available documentation
  + Interactive tutorials made by Google
* Who uses Go
  + Google
  + Uber
  + Twitch
* Syntax
  + Syntactically like C
  + Includes Java oriented safety
  + Variables
    - Statistically typed and bound at compile time
  + = normal assignment operator
  + := quick assignment operator that creates dynamically typed language
* Difficult to install and setup runtime environment for IDE
  + Easy to set up IDE support
    - Eclipse plugin

Lua

* 1993
  + Roberto Lerusalimschy
  + Luiz Henrique de Figueriedo
  + Waldemar Celes
* Purpose
  + Multi-paradigm scripting language with support for other types of programming paradigms
  + Influenced by
    - Modula- control structures
    - CLU- multiple assignments and returns from function calls
    - C++- being able to declare a local variable where it was needed
    - Scheme- list and table as primary data structures
  + Functionality
    - Fast, benchmarked as one of the fastest interpreted scripting languages
    - Portable
    - Embedded
    - Powerful
    - Small- 24000 lines of C
    - Free/open source
    - Garbage Collection
    - Dynamically typed, automatic memory management
  + Syntax
    - Expressions and declarations do not need to end with a character, such as ; or .
    - Variables cannot start with numbers
    - Dynamically typed
    - Can redefine with different types
    - Only one numeric type “number”
    - Variables default to nil
    - Case sensitive
    - Not whitespace sensitive
    - Table indices start at 1, not 0
    - Comments begin with –
* Tables and Keys
  + Tables are only container type in Lua
  + Tables are associative arrays with key/value pairs
  + Keys can be assigned any value
* Metatables- table with defines behaviors of operations on tables
  + Tables and Metatables can be constructed in a manner to build object/class-like structures in Lua
* Uses
  + Game Engines
  + TeamSpeak
  + VLC Scripting
  + Adobe Photoshop
  + Aerospike Database

Rust

* Graydon Hoare
* Began in 2006, Mozilla sponsored in 2009, launched 2010
* LLVM as its back-end
* Not considered “original” since its its design elements come from a wide range of sources
* Influenced by C++, etc.
* Designed to help developers create fast and secure applications
* Designed to prevent problems like seg fault and guaranteeing thread safety
* Addressed zero-cast abstractions
* Who uses Rust
  + Mozilla
  + Npm
  + Dropbox
  + Yelp
  + Sentry
  + Tilde
* Functionality
  + Multi-paradigm
  + Emphasis on safety, control of memory and easy concurrency
  + Statically typed, but types can be inferred by compiler
  + Goal is achieved by its implementation of ownership
  + Many features that can be OO, but also some OO features that it does not have
  + Can define struct and implementation blocks to define data
  + Does NOT have implementation inheritance, but does support interface inheritance
  + Traits can be used for encapsulation
* Compiler by default uses static linking
* Executables contain every dependency so compatible systems need nothing other than the executable to run
* Key Features
  + Ownership- values have unique owners with sharing scope with owner
  + Borrowing- there can be multiple immutable references
  + Lifetimes- prefers stack allocation, can be declared in code
  + Compiler- type inference checked at compile time; numerous warnings and errors
* Package Manager
  + Native package manager is called Cargo
    - Downloads packages and dependencies for compiling projects
    - Uploads packages to crates.io
* Syntax
  + fn main()
  + fn()
  + let- binds variable
  + let mut- makes a variables that is a binded mutable
  + struct
  + enum
  + println!()
  + pub
  + match
  + static
* Summary
  + Easy to learn and read
  + Comparable to C/C++
  + Great standard library
  + Easy syntax
  + Can cross compile with other editors
  + Still growing