## Dynamically Typed Languages:

* A language can be considered dynamically typed when the variable type is associated by the runtime values of the system not named fields.
* Dynamically typed languages are often less verbose
* With dynamically typed languages, you don’t need to wait until a compiler is finished before changes can be tested
* Debugging time is spent mostly on logic errors rather than syntax issues
* When refactoring the code, the area in which they affect is less

## Statically typed languages

* A language would be considered to be statically typed if the type of variables being used is known to the system at the time in which it is complied.
* A large amount of checking can be done with the compiler, this can catch some of the more trivial issues at an earlier stage.
* Can work with relational databases easier than a dynamically typed language would be able to.
* There are more options with dynamically typed languages for compiler optimization