编程语言—抽象加规则—底层语言（汇编和机器）

由于抽象的角度和规则制定的不同导致编程语言的语法和抽象程度也不一样。

**变量：**

区分大小写，类型，定义和声明，作用域（c中有无static或者像javascript中的有无var声明的变量其作用域是不一样的）

**作用域规则：**

**函数：**

声明：参数传递是按值传递还是按引用传递，参数列表是否可变，是否需要声明返回类型，是否支持嵌套，参数是否需要声明类型

调用：

**字符串：**

**类型系统：**

基本类型（简单的值）和非基本类型（对象）

静态类型还是动态类型

强类型还是弱类型

怎样进行类型转换

对于动态类型一般会有类型检测机制

**表达式、语句以及表达式语句：**

表达式类型：值表达式（最简单的表达式）、一元表达式、二元表达式等

表达式可以出现的地方有哪些：赋值语句

**值：**

是原始的值还是对象

如果有原始的值，则原始的值是否有相应的装箱函数将其转换成对象，使得有相应的方法可以对数据进行操作。

**控制结构：**

分支语句：if/else switch（多分支，swich支持的值是否只能为整型还是能够支持其它类型的值）

循环:

While do/while until

For

Foreach

对对象的键进行迭代的for结构，如for..of

对对象的值进行迭代的for结构，如for ..in

**块结构：**

像表达if/while/for等控制结构以及函数、类等块结构是使用{}还是end形式（其中if..then..fi也算是end结构吧）

面向对象：

基于类、原型、元表等

实现多继承方式：接口（java）、mixin（ruby）、C3序列化（python）

**垃圾收集机制：**

**错误处理：**

异常机制检测型异常（java）