

Naming (I)

Overview

- A critical ability of programming languages.
 - variables, functions, and programs need names before defining them.
- A name implies a **binding**, which is the connection between a **definable entity** and an **identifier**.
 - A binding is **static** if it takes place **before run time**.
 - Example: direct C function call. The function referenced by the identifier cannot change at run time
 - A binding is **dynamic** if it takes place **during run time**.
 - Example: non-static methods in Java (methods that can be overridden or inherited). The specific type of a polymorphic object is not known before run time (in general), the executed function is dynamically bound.
- Names (identifiers) are **determined by lexical rules**.
 - Case sensitivity

Case-insensitive	HTML, VHDL
Case-Sensitive	C, Java, Python
Mixed case sensitivity	PHP, MySQL

- In PHP, variables are case sensitive, but functions are not.
- MySQL provides an option for case-sensitive table names, and the query sentences are case-insensitive.
- Use of special characters: some languages allow use special characters

	Allowed	Not Recommend
Underscore (_)	C, Java, Python	SQL (considered as alias)

- Predefined identifiers
 - Reserved: **keywords/reserved** words
 - Most of PLs do not allow redefine the keywords. This is because it can make the parsing more efficiently.
 - For example, when parsing a program, the keyword, if, is always the start of a conditional statement. But if the keyword is redefined, it would need a second or more round to decide the meaning of the keyword.