# Types of Identifiers

<table>
<thead>
<tr>
<th>מ限りים בשפה</th>
<th>מérioוח השוק</th>
<th>שימור מספר</th>
<th>דוגמה</th>
<th>סוג</th>
</tr>
</thead>
<tbody>
<tr>
<td>כן</td>
<td>לא</td>
<td>כן</td>
<td>print AWK</td>
<td>מוגדר שומר בשפה AWK</td>
</tr>
<tr>
<td>כן</td>
<td>לא</td>
<td>לא</td>
<td>WriteLn</td>
<td>מוגדר בשפת פסקל</td>
</tr>
<tr>
<td>כן</td>
<td>לא</td>
<td>לא</td>
<td>printf C</td>
<td>מוגדר בשפת C</td>
</tr>
<tr>
<td>לא</td>
<td>כן</td>
<td>לא</td>
<td>HelloWorld</td>
<td>המוגדר בשפת פסקל</td>
</tr>
</tbody>
</table>

*Introduction*
Hello World in Ada

-- Hello World in Ada

with Text_IO;

procedure Hello_World is
begin
  Text_IO.Put_Line(“Hello World!”);
end Hello_World;

• ‘Put_Line’ is an identifier defined in the library ‘Text_IO’
• The library is imported through the keyword ‘with’
Hello World in Java

// Hello.java: my first Java program. It prints // the string "Hello World" on the standard output // stream, and terminates.

class HelloWorld {
    static public void main(String args[]) {
        System.out.println("Hello World!");
    }
}

The library is imported automatically (by the compiler)
Symbols for Denoting Blocks

- The {} family of languages
  - Java, C++, C#, AWK, GO, ...
- No delimiters, see Occam program (next foil)
- begin… end (e.g., Pascal)
  - Attaching a reference to the starting point
    - e.g., ‘end Hello_World’ in Ada
- if… fi, case… esac (Algol 68)
Hello World in Occam

-- Hello world in Occam
#INCLUDE "hostio.inc"
#USE "hostio.lib"
PROC hello.world (CHAN OF SP fs, ts)
SEQ
  so.write.string.nl(fs, ts, "Hello World!")
SEQ i = 1 FOR 10
SEQ
  so.write.int(fs, ts, i, 0)
  so.write.nl(fs, ts)
Hello World in Eiffel

```eiffel
note "Hello World in Eiffel"
class HELLO
create run
feature run
  do
    print ("Hello World!\n")
  end
end HELLO
```

- Notice the use of the keyword ‘note’ to indicate a one-line comment
Development Environments and The Program Starting Point

- The program code refers to the code written by the programmers (user code) and a set of imported (standard) libraries
- Defining which files constitute a program and the starting point
  - Autarchic approach
  - Metaphysic approach
  - Holistic approach
Autarchic approach

- One of the keywords denotes the starting point
  - The keyword ‘program’ in Pascal
  - The keyword ‘BEGIN’ in AWK

- A program in Pascal and AWK
  - contains one single file
  - No ‘standard libraries’
Metaphysic Approach

- The program starts from a function with a specific name
  - The name is not a keyword
  - The starting point is not part of the language definition

- Most implementations of C use the name ‘main’ to denote the starting function
- The implementation of C for Windows uses the name ‘WinMain’
- The compiler denotes which files constitute the program
C for MS-Windows

/* Hello world in C for MS-Windows */

#include <windows.h>

int PASCAL WinMain(HINSTANCE hInstance, 
                   HINSTANCE hPrevInstance, LPSTR CmdLine, int Show)
{
    MessageBox( 
        GetActiveWindow(), 
        "Hello World!",
        "Hello Windows World",
        MB_OK);
    return 0;
}
Holistic Approach

- Not part of the grammar
- Yet, the language defines how to identify the files that constitute the program and what is the starting point
  - These definitions are independent on the development environments (platforms)
- Eiffel (through the Cluster file), Java
String Literals

- Different ways to denote string literals
  - ‘Hello World’
    - Cobol, Pascal
  - “Hello World”
    - C, Java, Ada, AWK, Eiffel, Cobol, Pascal…
  - (Hello World)
    - PostScript
  - 12H Hello World
    - Fortran IV

- Denoting the end-of-line as part of the literal vs. using different function to enforce new line
  - “Hello World\n” in C, Go
  - “Hello World%N” in Eiffel
  - WriteLn in Pascal; println in Java

- Escaping character (usually backslash)
  - Enables the use of the string delimiter as part of the string itself
  - Introducing characters that have no graphical representation

- Pascal uses a different technique
  - ‘Let’’s say “Let”’s”
  - “WriteLn(‘Hello World’)”

- String literal that spans several lines
  - The use of backslash in C as the last character in the line
  - Automatic concatenation of two consecutive string literals (in C)
  - Impossible in Java (use the ‘+’ operator to concatenate literals at run time)
More Topics

- Separators vs. Terminators
  - Separatistic grammar
  - Terministic grammar
  - Liberal grammar

- Comments
  - Line comment
  - Block comment
  - Nested comments
Program Execution Model

☐ A language design must be specific about how all basic operations are done
☐ Predict program running time
☐ Examples of execution models:
  ☐ Fortran:
    - Flat register machine
    - No stacks, no recursion
    - Memory arranged as linear array
  ☐ Algol family:
    - Stack of activation records
    - Heap storage: an area of memory used for dynamic memory allocation
  ☐ Smalltalk:
    - Objects, communicating by messages