

<i>C#</i>	<i>C</i>	<i>JAVA</i>	<i>JavaScript</i>	<i>PHP</i>	<i>Python</i>	<i>Ruby</i>
class figura{}	typedef struct {} figura	class figura{}	class figura{}	class figura{}	class figura:	class figura end
abstract class figura{}	No existe	abstract class figura {}	class figura2 extends figura{}	abstract class figura{}	class figura: @abc.abstractmethod	class figura < figura2 end
interface Ifigura{}	No existe	interface figura{}	Interface figura{}	interface iFigura{}	No existe palabra clave. Se utilizan clases abstractas	No existe. Se puede emular declarando un modulo
internal class figura{}	No existe	package figura	No existe	No existe	No existe	No existe
partial class figura{}	No existe	No existe	No existe. Se puede hacer por convenciones	No existe	No existe	No existe
private class figura{}	No existe	private class figura{}	class figura{#privateField}	No existe	No existe	private_constant :figura end
protected class figura{}	No existe	No existe	No existe	No existe	No existe	No existe
static class figura{}	No existe	class figura{static{}}	class figura{static figurar()}	No existe	class figura{@staticmethod}	No existe
If ('condicion') {} else {}	If('condicion') {} else {}	If('condicion') {} else {}	If('condicion') {} else {}	If('condicion') {} else {}	If('condicion'): else:	If'condicion' else: end
while ('condicion') {}	while ('condicion') {}	while ('condicion') {}	while ('condicion') {}	while ('condicion') {}	while ('condicion'):	while 'condicion' do end
for (i = 0, i < 10; i++) {}	for (i = 0, i < 10; i++) {}	for (i = 0, i < 10; i++) {}	for (i = 0, i < 10; i++) {}	for (i = 0, i < 10; i++) {}	for x in y:	for a in 1..5 do end
foreach (int i in nros) {}	No existe	for (int variable :lista) {}	variable.foreach ();	foreach (\$variable as \$variable) {} 	for x in y:	variable.each do end