Tabla de símbolos del lenguaje Ruby

La sintaxis de Ruby es similar a la de Perl o Python. La definición de clases y métodos está definida por palabras clave. Los operadores son un componente esencial de cualquier lenguaje de programación. Con ellos podemos asignar, unir, cambiar o comparar valores de datos, cambiar el flujo del programa, etc. Los operadores son símbolos que representan operaciones sobre un valor. Estos operadores requieren de dos operandos.

Palabra Reservada	Función
alias	Crea un alias para un operador, método o variable global que ya exista.
and	Operador lógico, igual a && pero con menor precedencia.
break	Finaliza un *while* o un *untilloop*, o un método dentro de un bloque
case	Compara una expresión con una clausula *when* correspondiente
class	Define una clase; se cierra con *end*.
def	Inicia la definición de un método; se cierra con *end*.
defined?	Determina si un método, una variable o un bloque existe.
do	Comienza un bloque; se cierra con *end*.
else	Ejecuta el código que continua si la condición previa no es *true*. Funciona con *if*, *elsif*, *unless* o case.

elsif	Ejecuta el código que continua si la condicional previa no es *true*. Funciona con *if* o *elsif*.
end	Finaliza un bloque de código.
ensure	Ejecuta la terminación de un bloque. Se usa detrás del ultimo *rescue*.
false	Lógico o Booleano *false*.
true	Lógico o Booleano *true*.
for	Comienza un loop *for*. Se usa con *in*.
if	Ejecuta un bloque de código si la declaración condicional es *true*. Se cierra con *end*.
in	Usado con el loop *for*.
module	Define un modulo. Se cierra con *end*.
next	Salta al punto inmediatamente después de la evaluación del loop condicional
nil	Vacio, no inicializado, invalido. No es igual a cero.
not	Operador lógico, igual como !.
or	Operador lógico, igual a // pero con menor precedencia.
redo	Salta después de un loop condicional.
rescue	Evalua una expresión después de una excepción es alzada. Usada después de *ensure*.
retry	Cuando es llamada fuera de *rescue*, repite una llamada a método. Dentro de *rescue* salta a un bloque superior.
return	Regresa un valor de un método o un bloque.
self	Objeto contemporáneo. Alude al objeto mismo.
super	Llamada a método del mismo nombre en la superclase.
then	Separador usado con *if*, *unless*, *when*, *case*, y *rescue*.
undef	Crea un método indefinido en la clase contemporánea.
unless	Ejecuta un bloque de código si la declaración condicional es *false*.
until	Ejecuta un bloque de código mientras la declaración condicional es *false*.
when	Inicia una clausula debajo de *under*.
while	Ejecuta un bloque de código mientras la declaración condicional es *true*.
yield	Ejecuta un bloque pasado a un método.
FILE	Nombre del archivo de origen contemporáneo.
LINE	Numero de la linea contemporánea en el archivo de origen contemporáneo.

Tabla de operadores con precedencia de más alta a más baja:

Método*	Operador	Descripción
SI	[ ] [ ]=	
SI	**	Exponente
SI	! ~ + -	Not, complemento, más y menos unarios
SI	* / %	Multiplicación, división, módulo
SI	Más, menos	
SI	>><<	Shift a la derecha e izquierda
SI	&	Bitwise And
SI		Bitwise Or y Or regular
SI	<= <>>=	Operadores de comparación
SI	<=> == === != =~ !~	Operadores de igualdad y coincidencia de patrones
	&&	And lógico
	ll	Or lógico
	.. ...	Rango incluyente y excluyente
	? :	If-then-else ternario
	= %= ~= /= = += |= &= >>= <<= *= &&= ||= **=	Asignación
	defined?	Revisar si un símbolo está definido

	not	Negación lógica
	or and	Composición lógica
	Ifunlesswhileuntil	Modificadores de expresión
	beginend	Expresiones de bloque

CARACTERES ESPECIALES	FUNCION
.	cualquier caracter
[]	especificación por rango. P.ej: [a-z], una letra de la a, a la z
\w	letra o número; es lo mismo que [0-9A-Za-z]
\W	cualquier carácter que no sea letra o número
\s	carácter de espacio; es lo mismo que [\t\n\r\f]
\S	cualquier carácter que no sea de espacio
\d	número; lo mismo que [0-9]
\D	cualquier carácter que no sea un número
\b	retroceso (0x08), si está dentro de un rango
\b	límite de palabra, si NO está dentro de un rango
\B	no límite de palabra
\t	Representa un tabulador.
\A	Representa el inicio de la cadena. No un carácter sino una posición.
\r	Representa el "retorno de carro" o "regreso al inicio" o sea el lugar en que la línea vuelve a iniciar.

\Z	Representa el final de la cadena. No un carácter sino una posición
\n	Representa la "nueva línea" el carácter por medio del cual una línea da inicio.
\f	Representa un salto de página
*****	cero o más repeticiones de lo que le precede
+	una o más repeticiones de lo que le precede
$	fin de la línea
{m,n}	como menos m, y como mucho n repeticioes de lo que le precede
?	al menos una repetición de lo que le precede; lo mismo que {0,1}
()	agrupar expresiones
||	operador lógico O, busca lo de antes o lo después
^	individual: representa el inicio de la en conjunto de los caracteres especiales "^" realiza validaciones en forma sencilla.
|	indicar una de varias opciones.

OPERADORES	FUNCION
:	Alcance (scope)
[]	Índices
******	Exponentes
+ - ! ~	Unarios: pos/neg, no,…
*** / %**	Multiplicación, División,…
+ -	Suma, Resta,…
« »	Desplazadores binarios,…
&	'y' binario
|, ^	'or' y 'xor' binarios
>>= <<=	Comparadores
== === <=> != =~ !~	Igualadad, inegualdad,…
&&	'y' booleano
||	'o' booleano
.. …	Operadores de rango
= (+=, -=,...)	Asignadores
?:	Decisión ternaria
not	'no' booleano
and, or	'y', 'o' booleano