JAVASCRIPT

edix Digital Workers

n Number() = 42

METHODS

- s .toExponential(dec) exp. notation
- s .toFixed(dec) fixed-point notation
- s .toPrecision(p) change precision
- b .isFinite(n) check if number is finite
- b .isInteger(n) check if number is int.
- b .isNaN(n) check if number is NaN
- n .parseInt(s, radix) string to integer
- n .parseFloat(s, radix) string to float

r Regexp() = /.+/ig

METHODS

- a .exec(str) exec search for a match
- b .test(str) check if regexp match w/str

CLASSES

. any character	\t tabulator
\d digit [0-9]	\r carriage return
\D no digit [^0-9]	\n line feed
\w any alphanumeric char [A-Za-z0-9_]	
\W no alphanumeric char [^A-Za-z0-9_]	
\s any space char (space, tab, enter...)	
\S no space char (space, tab, enter...)	
\xN char with code N	[\b] backspace
\uN char with unicode N	\0 NUL char

CHARACTER SETS OR ALTERNATION

[abc] match any character set
[^abc] match any char. set not enclosed
a|b match a or b

BOUNDARIES

^ begin of input	$ end of input

\b zero-width word boundary
\B zero-width non-word boundary

GROUPING

(x) capture group (?:x) no capture group
\n reference to group n captured

QUANTIFIERS

x* preceding x 0 or more times {0,}
x+ preceding x 1 or more times {1,}
x? preceding x 0 or 1 times {0,1}
x{n} n ocurrences of x
x{n,} at least n ocurrences of x
x{n,m} between n & m ocurrences of x

ASSERTIONS

x(?=y) x (only if x is followed by y)
x(?!y) x (only if x is not followed by y)

window = global interaction func.

METHODS

USER INTERACTION METHODS

- .alert(str) show message (ok button)
- s .prompt(str, def) ask answer to user
- b .confirm(str) show message (ok, cancel)

s String() = 'text'

PROPERTIES

- n .length string size

METHODS

- s .charAt(index) char at position [i]
- n .charCodeAt(index) unicode at pos.
- s .fromCharCode(n1, n2...) code to char
- s .concat(str1, str2...) combine text +
- b .startsWith(str, size) check beginning
- b .endsWith(str, size) check ending
- b .includes(str, from) include substring?
- n .indexOf(str, from) find substr index
- n .lastIndexOf(str, from) find from end
- n .search(regex) search & return index
- n .localeCompare(str, locale, options)
- a .match(regex) matches against string
- s .repeat(n) repeat string n times
- s .replace(str|regex, newstr|func)
- s .slice(ini, end) str between ini/end
- s .substr(ini, len) substr of len length
- s .substring(ini, end) substr fragment
- a .split(sep|regex, limit) divide string
- s .toLowerCase() string to lowercase
- s .toUpperCase() string to uppercase

onClick="..." (HTML) .onclick = (JS func) 'click' (Listener)

e events (only popular events)

MOUSE EVENTS

e .onClick	e .onDblClick
e .onMouseDown	e .onMouseUp
e .onMouseEnter	e .onMouseLeave
e .onMouseMove	e .onMouseOver
e .onMouseOut	e .onWheel

KEYBOARD EVENTS

e .onKeyDown	e .onKeyUp
e .onKeyPress	

```
.style.display
.style.backgroundColor
.style.color
.style.fontFamily
.style.fontSize
.style.opacity
.checked
.disabled
.value
.placeholder
.textContent
.focus()
.write(mensaje)
.getElementById(identificador)
.getElementsByName(identificador)
.addEventListener(evento,func);
```

JAVA SWING

edix Digital Workers

Componente	Método	Observaciones
JFrame	add(componente)	
	setTitle(cadena)	
	setLayout(organizador)	
	setResizable(bool)	
	setJMenuBar(JMenuBar)	
	setDefaultCloseOperation()	EXIT_ON_CLOSE DO_NOTHING_ON_CLOSE HIDE_ON_CLOSE DISPOSE_ON_CLOSE
	dispose()	
JLabel	setIcon(new ImageIcon(*ruta*))	
JComboBox	getSelectedItem()	
	getSelectedIndex()	
JCheckBox **JRadioButton**	isSelected()	
	setSelected(boolean)	
JMenuBar	add(JMenu)	
JMenu	add(JMenuItem)	
	addSeparator()	
componente	setForeground(Color)	Color.color new Color(r,g,b)
	setBackground(Color)	
	setFont(Font)	new Font(fuente, estilo, tamaño) Font.PLAIN BOLD ITALIC
	setEnabled(boolean)	
	requestFocus();	

Eventos	elemento.addActionListener(new ActionListener() {
	public void actionPerformed(ActionEvent e) {
	}
	});	

	elemento.addItemListener(new ItemListener() {
	public void itemStateChanged(ItemEvent e) {
	}
	});	

	elemento.addMouseListener(new MouseAdapter() {
	@Override
	public void mouseEntered(MouseEvent e) {
	}
	});	

JOptionPane	.showMessageDialog(padre, mensaje, titulo, tipo);	JOptionPane.*INFORMATION_MESSAGE* *ERROR_MESSAGE* *WARNING_MESSAGE* *QUESTION_MESSAGE* *PLAIN_MESSAGE*