



Manuel Matuzović

 @mmatuzo

ViennaJS 05/2017

# KEYBOARD CONTROLS

**FOR AN**

**INTERNATIONAL**

**USER BASE**

The image shows the interior of a bus, viewed from the back of the vehicle looking forward. The seats are arranged in rows on both sides of a central aisle. The ceiling has overhead handrails and lighting fixtures. Large windows on both sides provide a view of the outside world. The entire image is overlaid with a semi-transparent teal color. In the center, the name 'Manuel Matuzovic' is written in a large, white, sans-serif font.

**Manuel Matuzovic**

A map of Central Europe showing Austria, Hungary, Croatia, Bosnia and Herzegovina, and Serbia. A red location pin is placed in Slavonki Brod, Croatia. The map includes major cities like Munich, Vienna, Bratislava, Budapest, Zagreb, and Belgrade, as well as various roads and geographical features.

# Manuel Matuzović

**č, ć, đ, dž, š, ž, nj, lj**

**q, w, x**

# Keyboard layouts

***“A keyboard layout is any specific mechanical, visual, or functional arrangement of the keys, legends, or key-meaning associations (respectively) of a computer, typewriter, or other typographic keyboard.”***

# QWERTY

Esc	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	↑
± §	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	( 9	) 0	_ - [	+ =	Delete ⌫
Tab	Q	W	E	R	T	Y	U	I	O	P	{ [	} ]	Return ↵
Caps Lock ⬆	A	S	D	F	G	H	J	K	L	: ;	" '	 \ Enter	
Shift ⬆	~ `	Z	X	C	V	B	N	M	< ,	> .	? /	Shift ⬆	
Fn	Control	Option	Command					Command	Option		↑		
											←	↓	→

# QWERTZ

°	!	"	§	\$	%	&	/	(	)	=	?	'	←
^	1	2 <sup>2</sup>	3 <sup>3</sup>	4	5	6	7 {	8 [	9 ]	0 }	ß \	'	
↵	Q	W	E	R	T	Z	U	I	O	P	Ü	*	↵
	@		€									+ ~	
⇩	A	S	D	F	G	H	J	K	L	Ö	Ä	'	↵
												#	
⇧	>	Y	X	C	V	B	N	M	;	:	-	⇧	
	<							μ	,	.	-		
Strg	(Win)	Alt							Alt Gr	(Win)	(Menu)	Strg	

# AZERTY





# JSUKEN

Ё	!	"	№	;	%	:	?	* Р	( )	-	+	/	←
1	2	3	4	5	6	7	8	9	0	-	=	\	
Tab	Й	Ц	У	К	Е	Н	Г	Ш	Щ	З	Х	Ъ	Enter
↔													↵
Caps Lock	Ф	Ы	В	А	П	Р	О	Л	Д	Ж	Э		
⬆													
⇧ Shift	Я	Ч	С	М	И	Т	Ь	Б	Ю	,	.		⇧ Shift
Ctrl	Win Key	Alt						Alt	Win Key	Menu	Ctrl		

**Soooo?**

# Events

- **keydown**

The keydown event is fired when a key is pressed down.

- **keyup**

The keyup event is fired when a key is released.

- **keypress**

The keypress event is fired when a key is pressed down

# Values

- **keyCode**

Returns a numerical code identifying the value of the pressed key.

- **which**

Returns a numeric code identifying the value of the pressed key; this is usually the same as keyCode.

- **charCode**

Returns a Number representing the Unicode reference number of the key; this attribute is used only by the keypress event.

# event.keyCode

```
window.addEventListener('keydown', navigate);
```

```
function navigate(e) {  
  ...  
  
  if (e.keyCode === 68) {  
    moveLeftAndRight(1);  
  }  
  
  if (e.keyCode === 90) {  
    shootMissile();  
  }  
}
```

# Issues

- **Different meaning of properties** when handling a key event (keydown or keyup) versus a character event (keypress).
- The **values of keys** in keypress events are **different** between browsers.
- **Inconsistent values** for keys and events **cross-browser**
- keyCode on key events **tries to be international-friendly**, but isn't



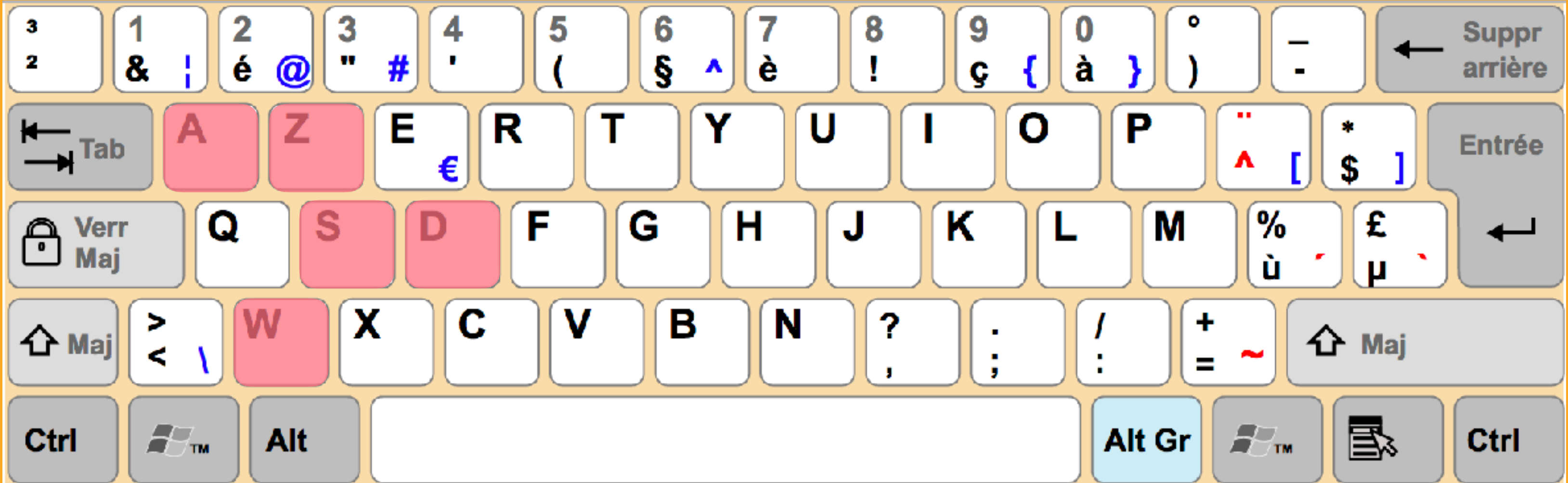




# QWERTY

Esc	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	↑
± §	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	( 9	) 0	_ - = +	Delete ⌫	
Tab	Q	W	E	R	T	Y	U	I	O	P	{ [	}	Return ↵
Caps Lock ⌵	A	S	D	F	G	H	J	K	L	: ;	" '	 \ Enter	
Shift ⬆	~ `	Z	X	C	V	B	N	M	< ,	> .	? /	Shift ⬆	
Fn	Control	Option	Command						Command	Option	⬆	⬆	⬆
											⬅	⬇	➡

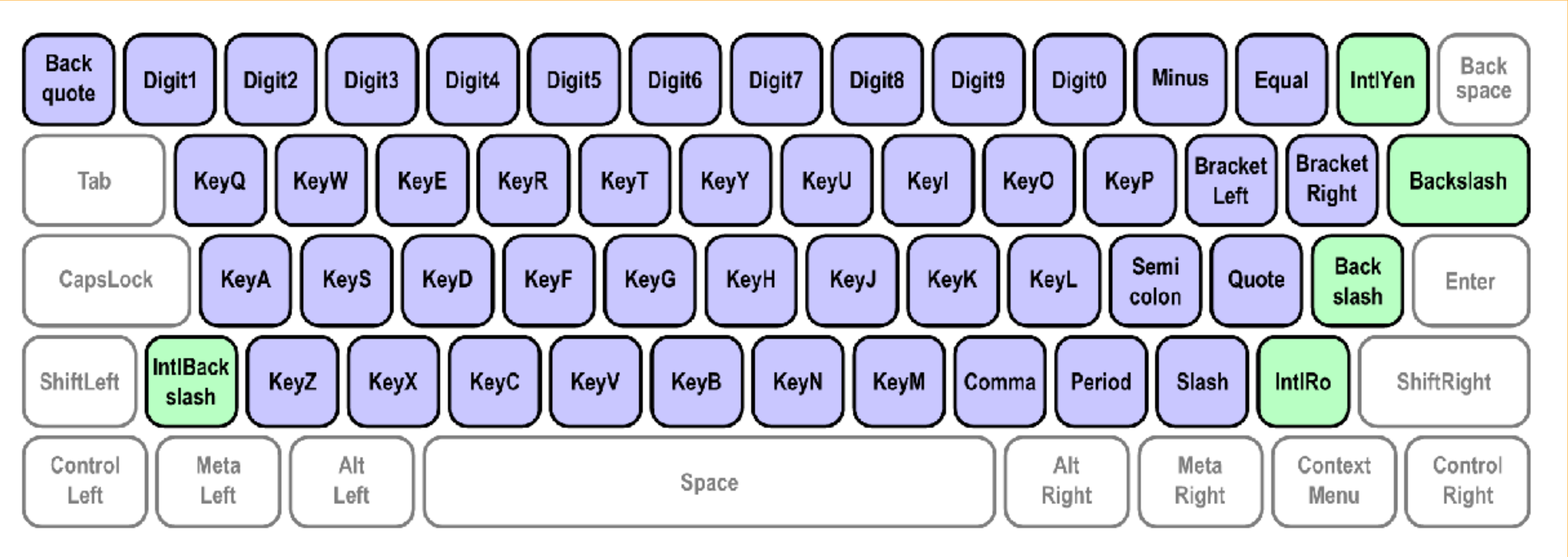
# Remember AZERTY?



# UI Events: API

- Two new properties: **key** and **code**
- **event.key** - printable character or a descriptive string, e.g. z
- **event.code** - physical key, e.g. KeyY
- Reference keyboard in the specification

# Reference keyboard



# event.keyCode

```
window.addEventListener('keydown', navigate);
```

```
function navigate(e) {  
  ...  
  
  if (e.code === 'KeyD') {  
    moveLeftAndRight(1);  
  }  
  
  if (e.code === 'KeyY') {  
    shootMissile();  
  }  
}
```



# Browsersupport

KeyboardEvent.key 📄 - WD

Global

63.45% + 5.62% = 69.07%

A `KeyboardEvent` property whose value is a string identifying the key that was pressed. Covers character keys, non-character keys (e.g. arrow keys), and dead keys.

Current aligned Usage relative Date relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			56			9.3		4.4	
	14	52	57	10		10.2		4.4.4	
11	15	53	58	10.1	44	10.3	all	56	57
		54	59	TP	45				
		55	60		46				
		56	61						

# Browsersupport

## KeyboardEvent.code 📄 - WD

Global

36.81%

A `KeyboardEvent` property representing the physical key that was pressed, ignoring the keyboard layout and ignoring whether any modifier keys were active.

Current aligned Usage relative Date relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			56			9.3		4.4	
	14	52	57	10		10.2		4.4.4	
11	15	53	58	10.1	44	10.3	all	56	57
		54	59	TP	45				
		55	60		46				
		56	61						



# Fallback

```
switch (e.code || e.key || e.keyCode) {  
  ...  
  
  case 'KeyD':  
  case 'ArrowRight':  
  case 39:  
    moveLeftAndRight(1);  
  ...  
}
```

# Notes

- There is no way of checking for the current keyboard layout.
- Use the **repeat** property to check if the user keeps pressing a key and the event is sent repeatedly
- Boolean properties for modifier keys (**altKey**, **ctrlKey**, **metaKey**, **shiftKey**).

HTML CSS JS

```

1 document.addEventListener('keyup', event_info);
2
3 document.addEventListener('keypress', event_info);
4
5 document.addEventListener('keydown', event_info);
6
7 function event_info(e) {
8   console.log('%c%s', 'background: #fff; color: #333;', e.type)
9   console.log(`which ${e.which}`)
10  console.log(`keyCode ${e.keyCode}`)
11  console.log(`repeat ${e.repeat}`)
12  console.log(`charCode ${e.charCode}`)
13  console.log(`code ${e.code}`)
14  console.log(`key ${e.key}`)
15  console.log(`altKey ${e.altKey}`)
16  console.log(`ctrlKey ${e.ctrlKey}`)
17  console.log(`metaKey ${e.metaKey}`)
18  console.log(`shiftKey ${e.shiftKey}`)
19 }

```

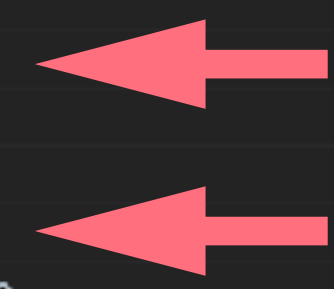
top Filter Info

```

keydown
which 91
keyCode 91
repeat false
charCode 0
code MetaLeft
key Meta
altKey false
ctrlKey false
metaKey true
shiftKey false
keydown
which 75
keyCode 75
repeat false
charCode 0
code KeyK
key k
altKey false
ctrlKey false
metaKey true
shiftKey false
keyup
which 91
keyCode 91
repeat false
charCode 0
code MetaLeft
key Meta
altKey false
ctrlKey false
metaKey false
shiftKey false

```

VM59 console runner-079c09a...js:1



<https://codepen.io/matuzo/pen/PmgWRm?editors=0010>

## TWEETABLE TAKE-AWAY

**The web is international and so are your users.  
Don't assume that all users use the same input  
devices.**



Manuel Matuzović

 @mmatuzo

ViennaJS 05/2017


[medium.com/@matuzo](https://medium.com/@matuzo)

[codepen.io/matuzo](https://codepen.io/matuzo)

[manuel@matuzo.at](mailto:manuel@matuzo.at)

[meetup.com/webclerks](https://meetup.com/webclerks)

**THANK YOU**

C  DEPENDEN  
MEETUP

JUNE 27TH 18:30 SEKTOR5 SHOW & TELL

# Images

- [http://www.daniel.at/images/pictures/\\_DSF2655.jpg](http://www.daniel.at/images/pictures/_DSF2655.jpg)
- [https://en.wikipedia.org/wiki/QWERTZ#/media/File:KB\\_Germany.svg](https://en.wikipedia.org/wiki/QWERTZ#/media/File:KB_Germany.svg)
- [https://en.wikipedia.org/wiki/QWERTY#/media/File:KB\\_Intl\\_English\\_Mac\\_-\\_Apple\\_Keyboard\\_\(MC184Z\).svg](https://en.wikipedia.org/wiki/QWERTY#/media/File:KB_Intl_English_Mac_-_Apple_Keyboard_(MC184Z).svg)
- [https://upload.wikimedia.org/wikipedia/commons/4/41/Belgian\\_pc\\_keyboard.svg](https://upload.wikimedia.org/wikipedia/commons/4/41/Belgian_pc_keyboard.svg)
- [https://upload.wikimedia.org/wikipedia/commons/6/60/KB\\_Russian.svg](https://upload.wikimedia.org/wikipedia/commons/6/60/KB_Russian.svg)