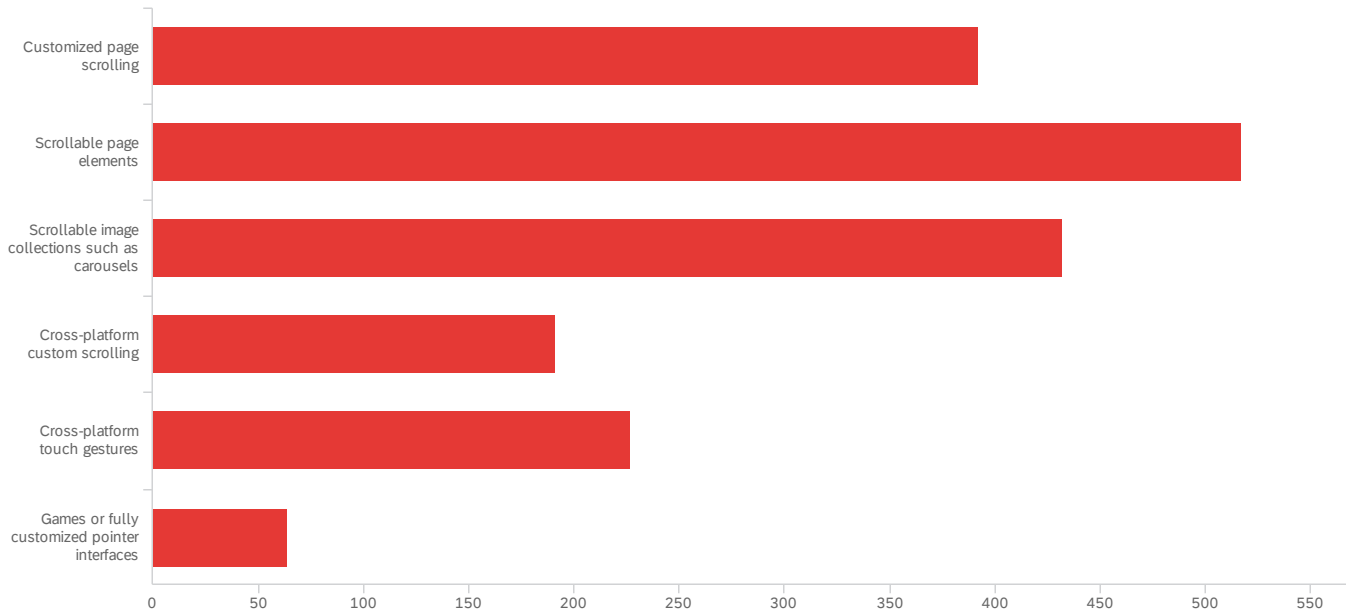


# Scrolling survey report

Scrolling features in CSS and JavaScript

May 5, 2021 3:44 AM MDT

## Q1 - What kinds of scrolling have you worked with on a production website?

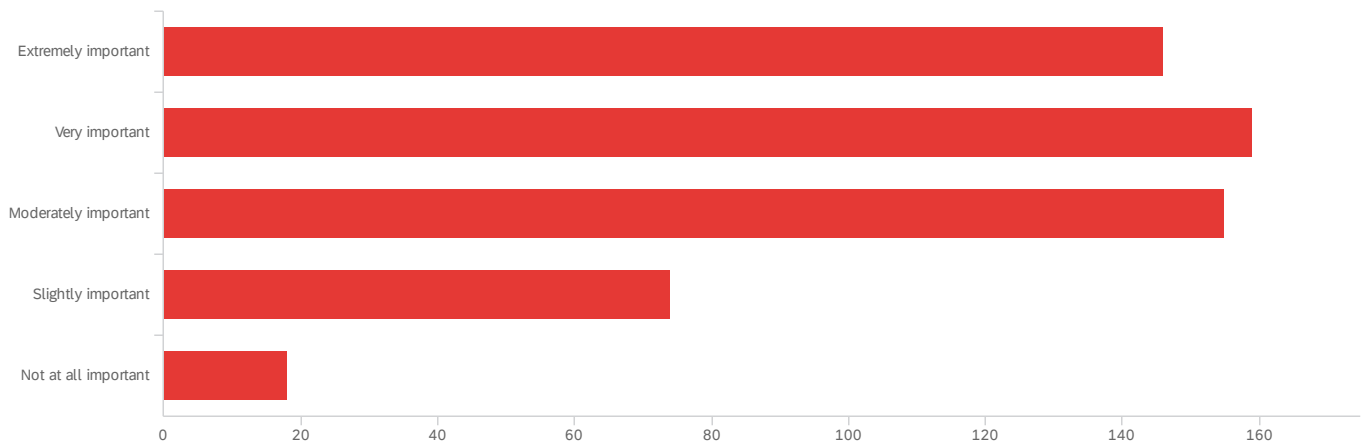


#	Field	Choice Count
1	Customized page scrolling	21.50% 392
2	Scrollable page elements	28.36% 517
3	Scrollable image collections such as carousels	23.70% 432
4	Cross-platform custom scrolling	10.48% 191
5	Cross-platform touch gestures	12.45% 227
6	Games or fully customized pointer interfaces	3.51% 64

1823

Showing rows 1 - 7 of 7

## Q2 - Are scrollable areas on web pages an important part of your work?

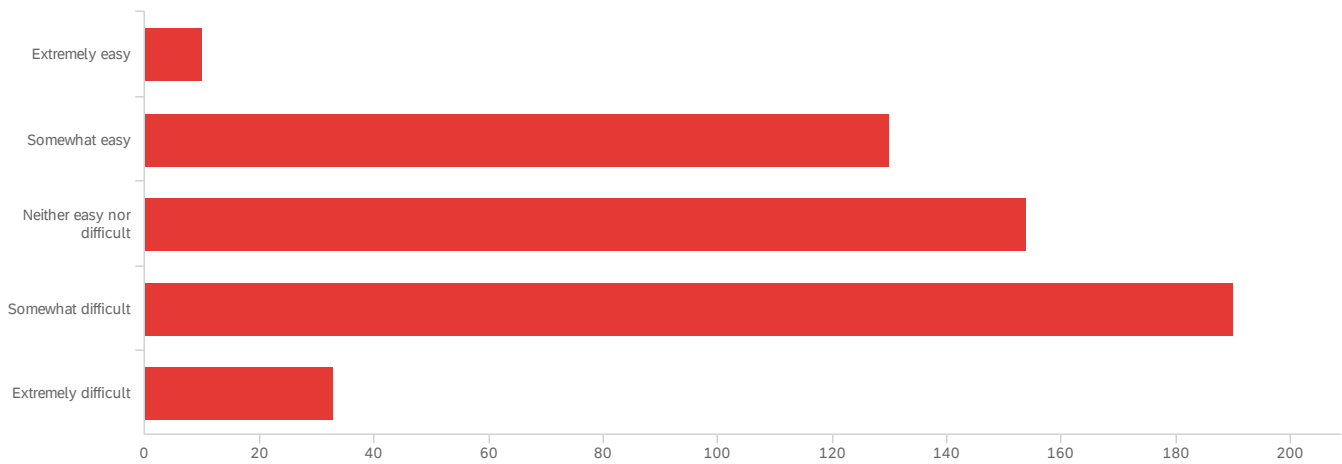


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Are scrollable areas on web pages an important part of your work?	1.00	5.00	2.38	1.11	1.23	552

#	Field	Choice Count
1	Extremely important	26.45% 146
2	Very important	28.80% 159
3	Moderately important	28.08% 155
4	Slightly important	13.41% 74
5	Not at all important	3.26% 18
		552

Showing rows 1 - 6 of 6

## Q2a - How would you rate working with scrollable areas?

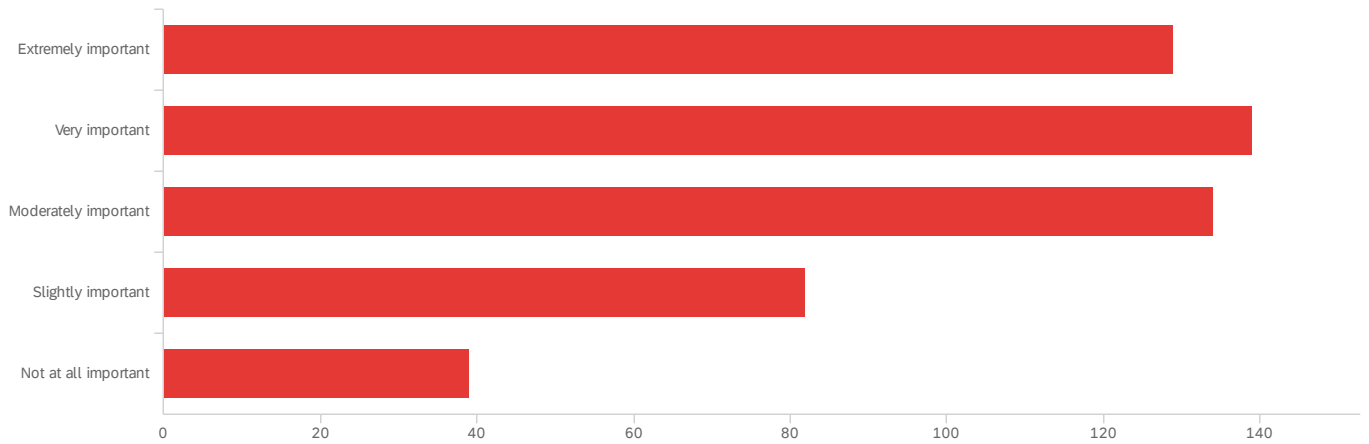


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How would you rate working with scrollable areas?	1.00	5.00	3.21	0.95	0.91	517

#	Field	Choice Count
1	Extremely easy	1.93% 10
2	Somewhat easy	25.15% 130
3	Neither easy nor difficult	29.79% 154
4	Somewhat difficult	36.75% 190
5	Extremely difficult	6.38% 33
		517

Showing rows 1 - 6 of 6

### Q3 - Are touchscreen interactions on web pages an important part of your work?

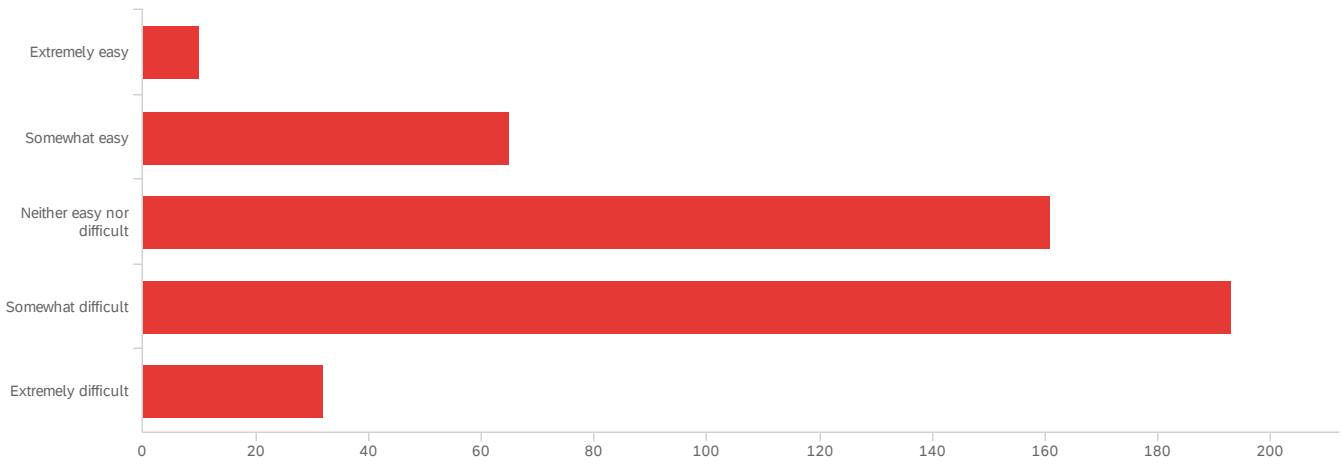


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Are touchscreen interactions on web pages an important part of your work?	1.00	5.00	2.55	1.23	1.50	523

#	Field	Choice Count
1	Extremely important	24.67% 129
2	Very important	26.58% 139
3	Moderately important	25.62% 134
4	Slightly important	15.68% 82
5	Not at all important	7.46% 39
		523

Showing rows 1 - 6 of 6

### Q3a - How would you rate working with touchscreen interactions?



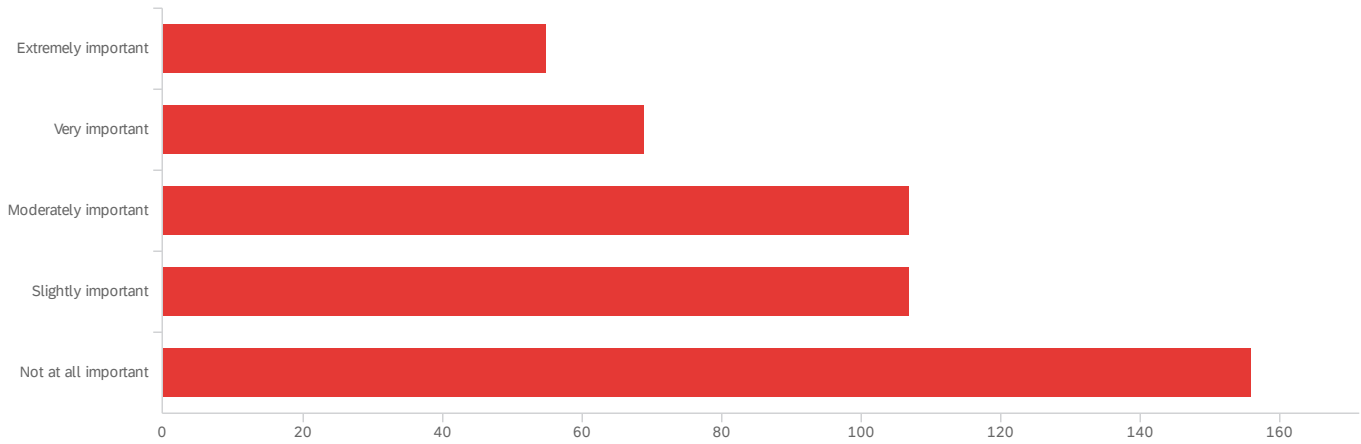
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How would you rate working with touchscreen interactions?	1.00	5.00	3.37	0.89	0.78	461

#	Field	Choice Count
1	Extremely easy	2.17% 10
2	Somewhat easy	14.10% 65
3	Neither easy nor difficult	34.92% 161
4	Somewhat difficult	41.87% 193
5	Extremely difficult	6.94% 32
		461

Showing rows 1 - 6 of 6

## Q4 - Are scroll-linked animations and interactions an important part of your work? For

example, scrolling of one element controls the display of another.

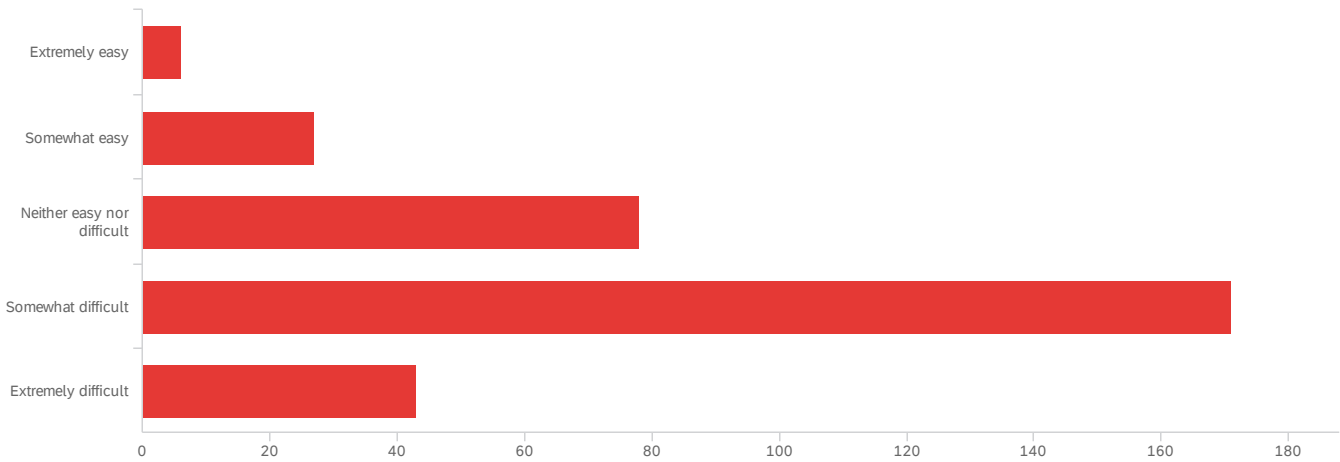


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Are scroll-linked animations and interactions an important part of your work? For example, scrolling of one element controls the display of another.	1.00	5.00	3.49	1.35	1.83	494

#	Field	Choice Count
1	Extremely important	11.13% 55
2	Very important	13.97% 69
3	Moderately important	21.66% 107
4	Slightly important	21.66% 107
5	Not at all important	31.58% 156
		494

Showing rows 1 - 6 of 6

## Q4a - How would you rate working with scroll linked animations and interactions?

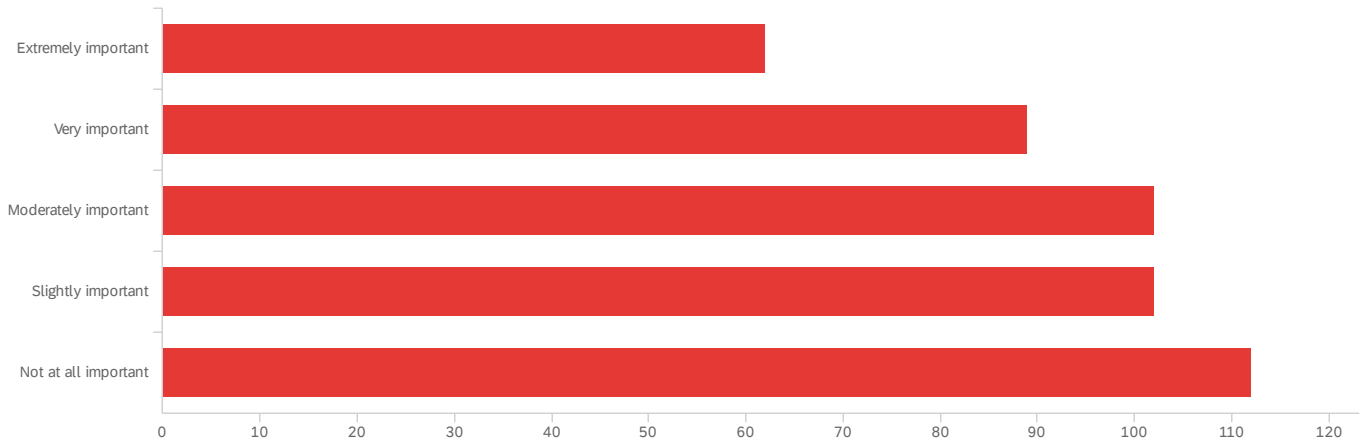


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How would you rate working with scroll linked animations and interactions?	1.00	5.00	3.67	0.87	0.76	325

#	Field	Choice Count
1	Extremely easy	1.85% 6
2	Somewhat easy	8.31% 27
3	Neither easy nor difficult	24.00% 78
4	Somewhat difficult	52.62% 171
5	Extremely difficult	13.23% 43
		325

Showing rows 1 - 6 of 6

Q5 - Are keyboard or gamepad scrolling UX important in your web work? For example, enabling gamepad buttons, or tab and shift-tab, to control navigation through a carousel.



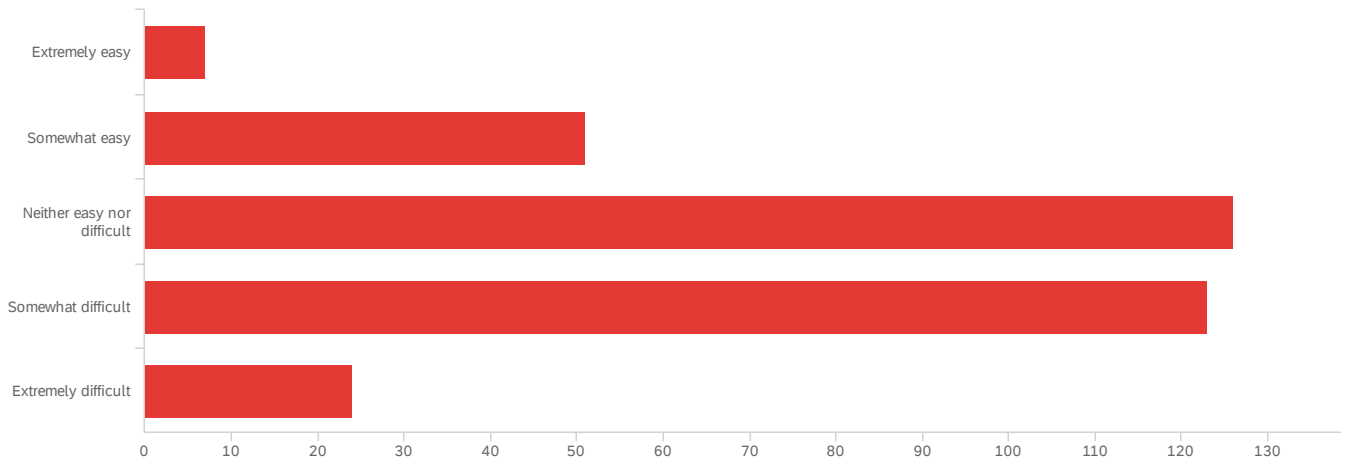
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Are keyboard or gamepad scrolling UX important in your web work? For example, enabling gamepad buttons, or tab and shift-tab, to control navigation through a carousel.	1.00	5.00	3.24	1.36	1.84	467

#	Field	Choice Count
1	Extremely important	13.28% 62
2	Very important	19.06% 89
3	Moderately important	21.84% 102
4	Slightly important	21.84% 102
5	Not at all important	23.98% 112
		467

Showing rows 1 - 6 of 6



## Q5a - How would you rate working with keyboard or gamepad scrolling?

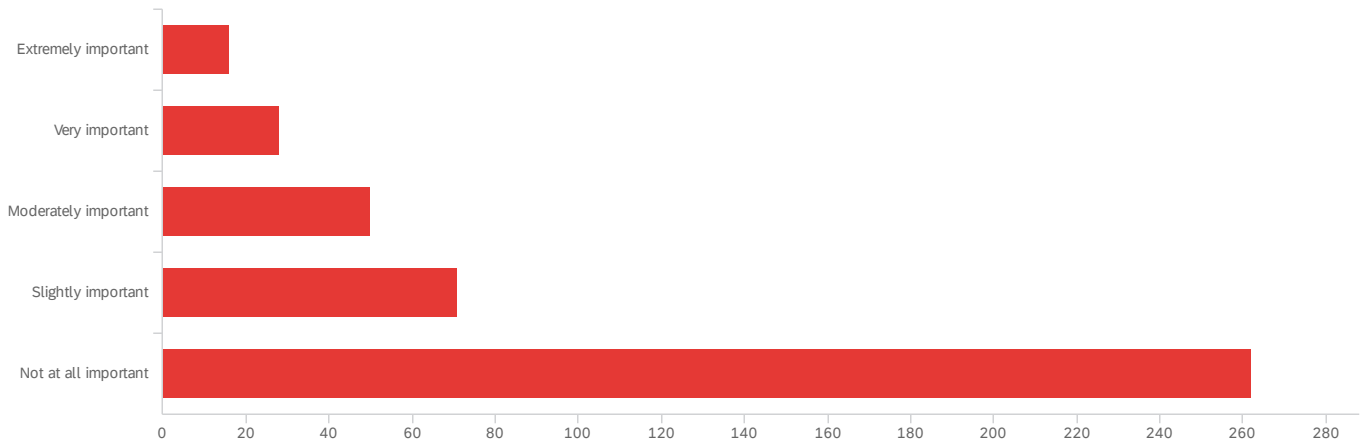


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How would you rate working with keyboard or gamepad scrolling?	1.00	5.00	3.32	0.89	0.80	331

#	Field	Choice Count
1	Extremely easy	2.11% 7
2	Somewhat easy	15.41% 51
3	Neither easy nor difficult	38.07% 126
4	Somewhat difficult	37.16% 123
5	Extremely difficult	7.25% 24
		331

Showing rows 1 - 6 of 6

## Q6 - Are international scroll directions important in your web work?

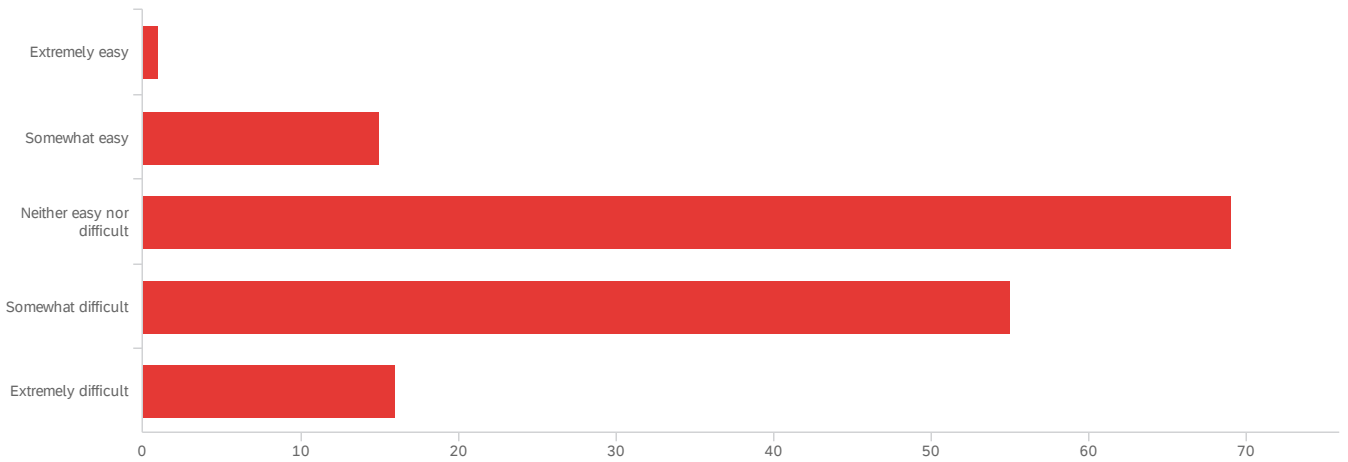


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	For	1.00	5.00	4.25	1.13	1.27	427

#	Field	Choice Count
1	Extremely important	3.75% 16
2	Very important	6.56% 28
3	Moderately important	11.71% 50
4	Slightly important	16.63% 71
5	Not at all important	61.36% 262
		427

Showing rows 1 - 6 of 6

## Q6a - How would you rate working with international scroll directions?

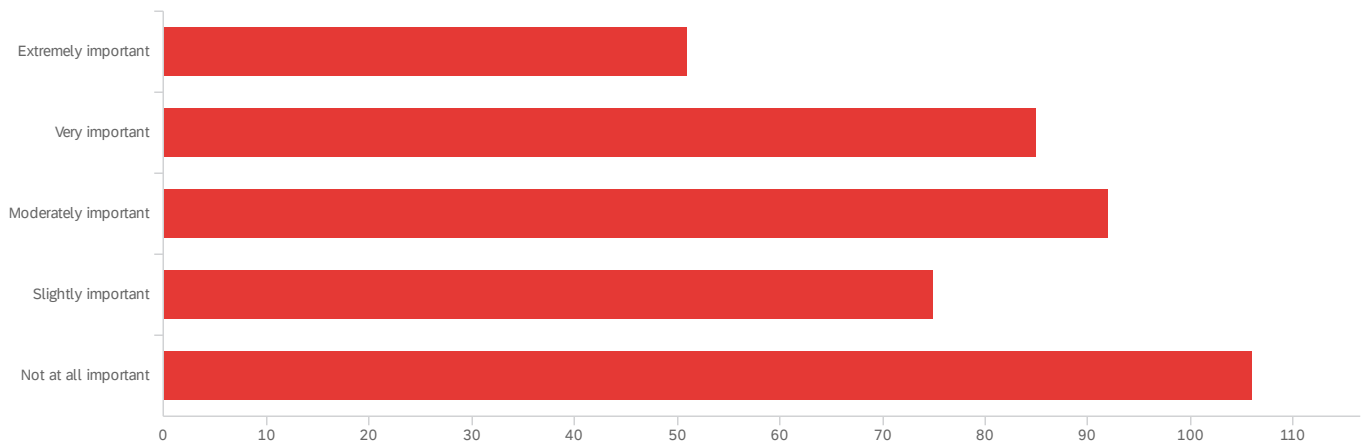


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How would you rate working with international scroll directions?	1.00	5.00	3.45	0.83	0.68	156

#	Field	Choice Count
1	Extremely easy	0.64% 1
2	Somewhat easy	9.62% 15
3	Neither easy nor difficult	44.23% 69
4	Somewhat difficult	35.26% 55
5	Extremely difficult	10.26% 16
		156

Showing rows 1 - 6 of 6

## Q7 - Is building scroll areas that overscroll like on native platforms important to you?



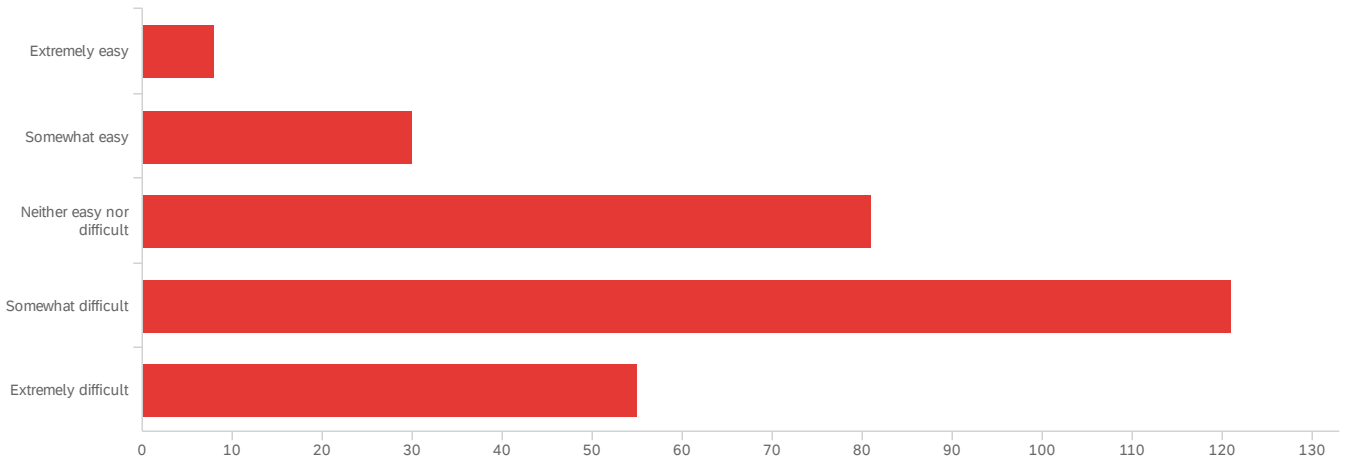
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Is building scroll areas that overscroll like on native platforms important to you? When a user has scrolled to the end of a scroll area, native platforms provide visual UI feedback to reinforce that the end of scrolling has occurred in that direction. On Android this is a cross-fading, finger tracking half circle. On MacOS and iOS it's an elastic bounce effect. These are overscroll styles.	1.00	5.00	3.24	1.37	1.87	409

#	Field	Choice Count
1	Extremely important	12.47% 51
2	Very important	20.78% 85
3	Moderately important	22.49% 92
4	Slightly important	18.34% 75
5	Not at all important	25.92% 106
		409

Showing rows 1 - 6 of 6

# Q7a - How would you rate working to ensure scroll areas overscroll like on native

platforms?

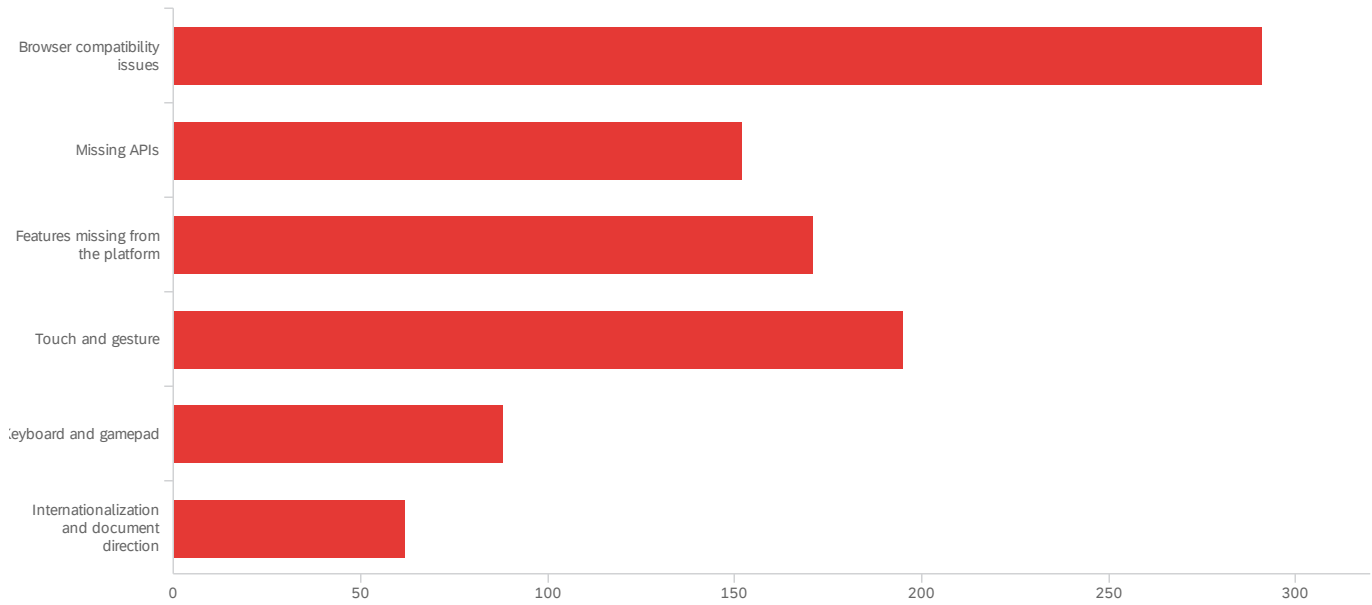


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How would you rate working to ensure scroll areas overscroll like on native platforms?	1.00	5.00	3.63	0.99	0.97	295

#	Field	Choice Count
1	Extremely easy	2.71% 8
2	Somewhat easy	10.17% 30
3	Neither easy nor difficult	27.46% 81
4	Somewhat difficult	41.02% 121
5	Extremely difficult	18.64% 55
		295

Showing rows 1 - 6 of 6

## Q8 - Do any of the following cause you difficulties when working with scrolling on the web?

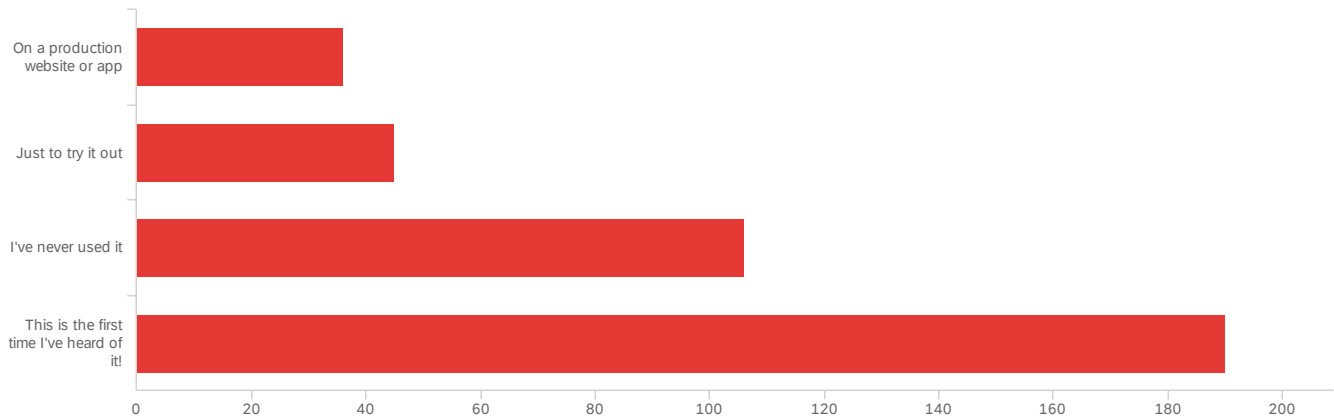


#	Field	Choice Count
1	Browser compatibility issues	30.34% 291
2	Missing APIs	15.85% 152
3	Features missing from the platform	17.83% 171
4	Touch and gesture	20.33% 195
5	Keyboard and gamepad	9.18% 88
6	Internationalization and document direction	6.47% 62

959

Showing rows 1 - 7 of 7

## Q9 - Have you used touch-action:manipulation?



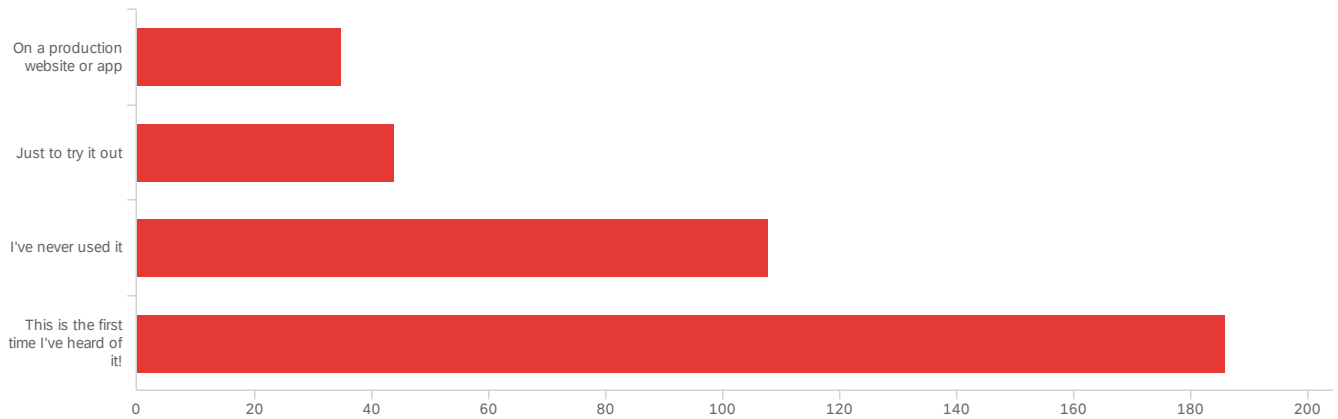
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	touch-action: manipulation enables double-taps to be disregarded and single-tap events to be dispatc	1.00	4.00	3.19	0.98	0.97	377

#	Field	Choice Count
1	On a production website or app	9.55% 36
2	Just to try it out	11.94% 45
3	I've never used it	28.12% 106
4	This is the first time I've heard of it!	50.40% 190

377

Showing rows 1 - 5 of 5

## Q10 - Have you used touch-action: pan-x pan-down?



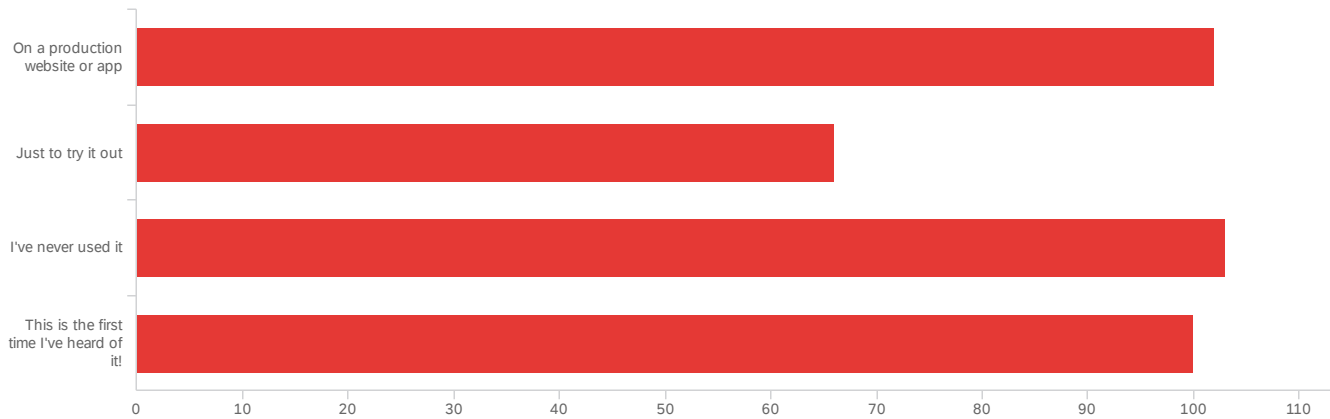
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	touch-action: manipulation enables double-taps to be disregarded and single-tap events to be dispatc	1.00	4.00	3.19	0.98	0.95	373

#	Field	Choice Count
1	On a production website or app	9.38% 35
2	Just to try it out	11.80% 44
3	I've never used it	28.95% 108
4	This is the first time I've heard of it!	49.87% 186
		373

Showing rows 1 - 5 of 5



## Q11 - Have you used overscroll-behavior: contain?



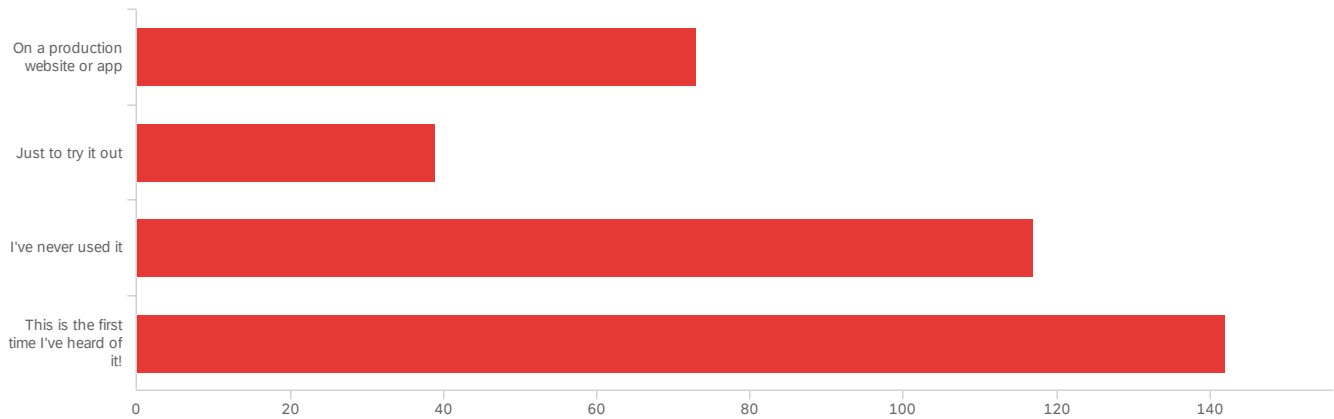
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	overscroll-behavior: contain Prevent a scroll container from leaking the scroll interaction into a parent scroller. Have you used this CSS property?	1.00	4.00	2.54	1.16	1.34	371

#	Field	Choice Count
1	On a production website or app	27.49% 102
2	Just to try it out	17.79% 66
3	I've never used it	27.76% 103
4	This is the first time I've heard of it!	26.95% 100

371

Showing rows 1 - 5 of 5

## Q12 - Have you used element.focus({preventScroll: true})?

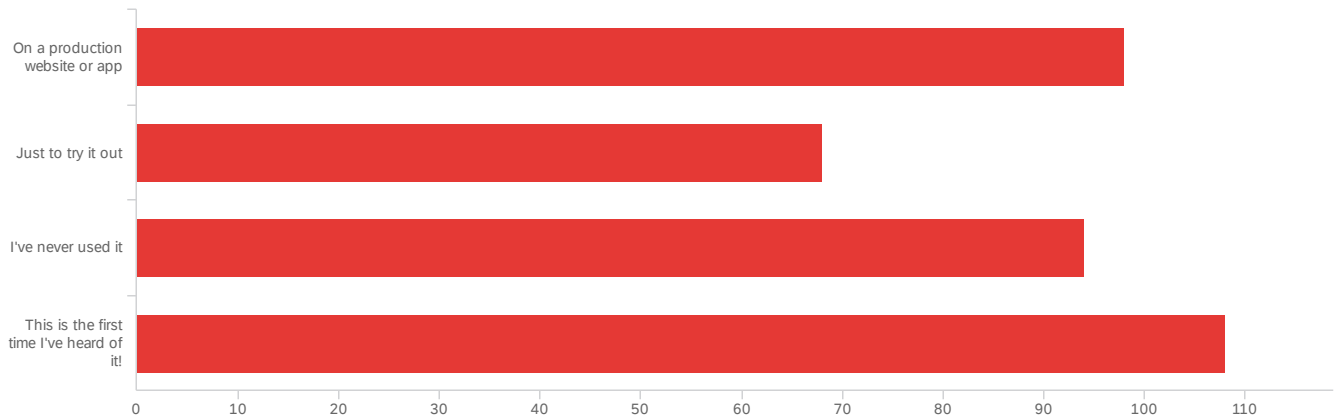


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	element.focus({preventScroll: true}) Prevent undesirable scroll results. Have you used this CSS property?	1.00	4.00	2.88	1.12	1.26	371

#	Field	Choice Count
1	On a production website or app	19.68% 73
2	Just to try it out	10.51% 39
3	I've never used it	31.54% 117
4	This is the first time I've heard of it!	38.27% 142
		371

Showing rows 1 - 5 of 5

## Q13 - Have you used scroll-padding or scroll-margin?



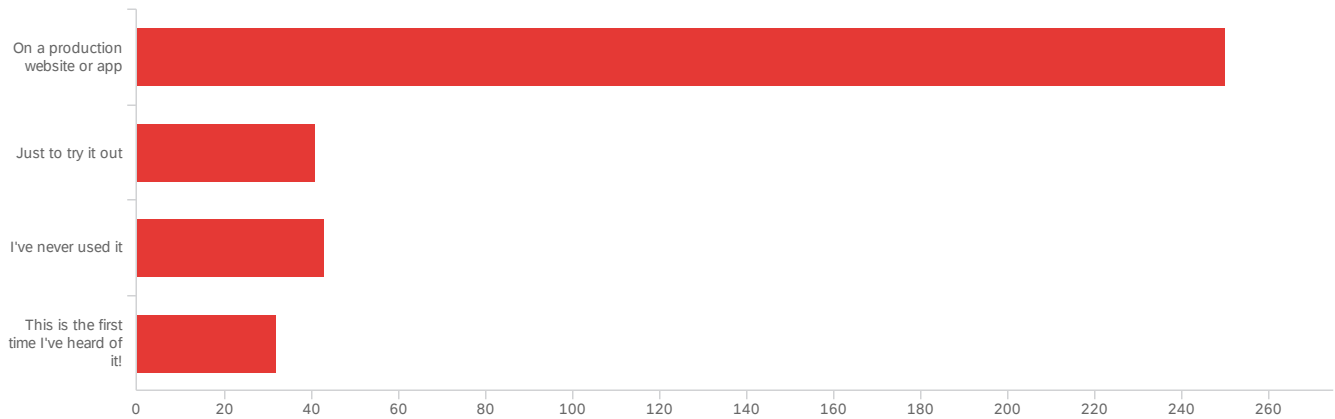
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	scroll-padding scroll-margin Offset an element from the viewport so it can be displayed optimally away from a header or navigation. Have you used these CSS properties?	1.00	4.00	2.58	1.17	1.36	368

#	Field	Choice Count
1	On a production website or app	26.63% 98
2	Just to try it out	18.48% 68
3	I've never used it	25.54% 94
4	This is the first time I've heard of it!	29.35% 108

368

Showing rows 1 - 5 of 5

## Q14 - Have you used scroll-behavior: smooth scrollIntoView({behavior: 'smooth'})?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	scroll-behavior: smooth scrollIntoView({behavior: 'smooth'}) Set the behavior for a scrolling area. Have you used this CSS property?	1.00	4.00	1.61	1.00	1.00	366

#	Field	Choice Count
1	On a production website or app	68.31% 250
2	Just to try it out	11.20% 41
3	I've never used it	11.75% 43
4	This is the first time I've heard of it!	8.74% 32

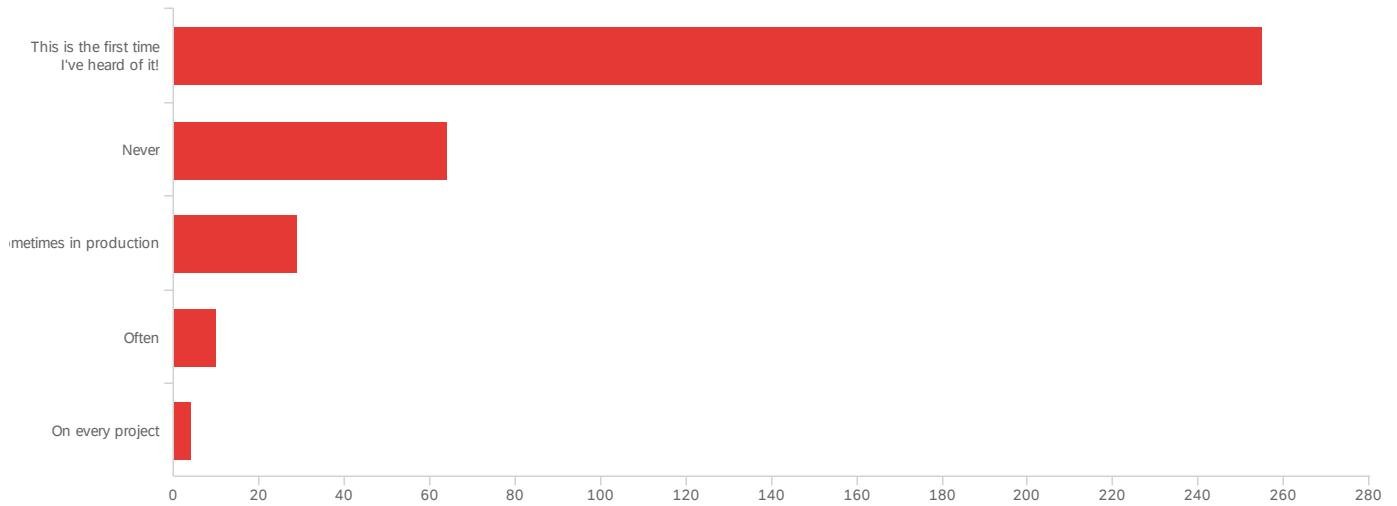
366

Showing rows 1 - 5 of 5

## Q15 - rootScroller Enable an arbitrary scrolling element to have special features

otherwise only available to the documentElement. Have you needed to be concerned with

the difference between the :rootScroller and other scrollers?

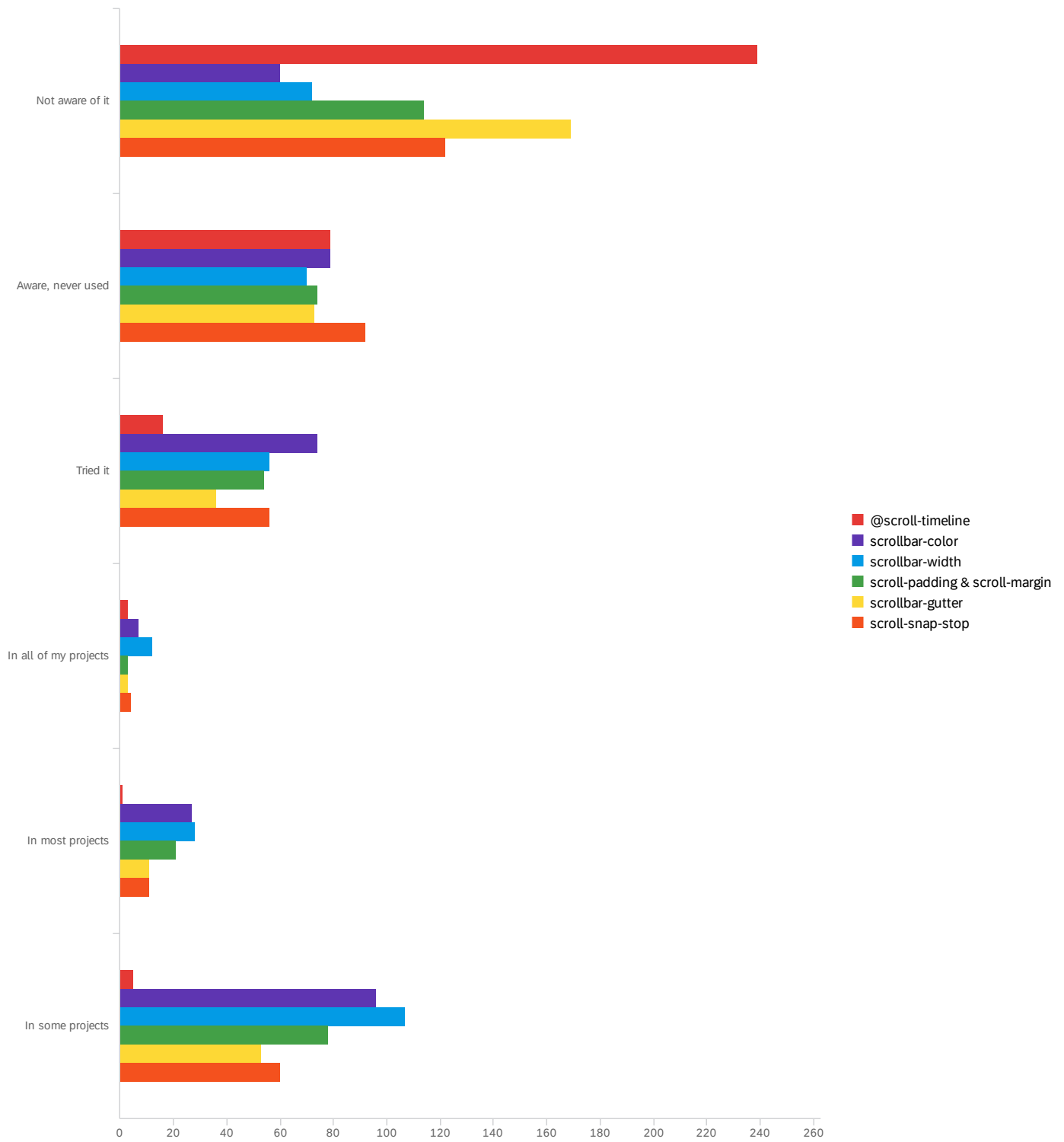


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	rootScroller Enable an arbitrary scrolling element to have special features otherwise only available to the documentElement. Have you needed to be concerned with the difference between the :rootScroller and other scrollers?	1.00	5.00	1.46	0.84	0.71	362

#	Field	Choice Count
1	This is the first time I've heard of it!	70.44% 255
2	Never	17.68% 64
3	Sometimes in production	8.01% 29
4	Often	2.76% 10
5	On every project	1.10% 4
		362

Showing rows 1 - 6 of 6

# Q16 - Have you used CSS scroll properties?



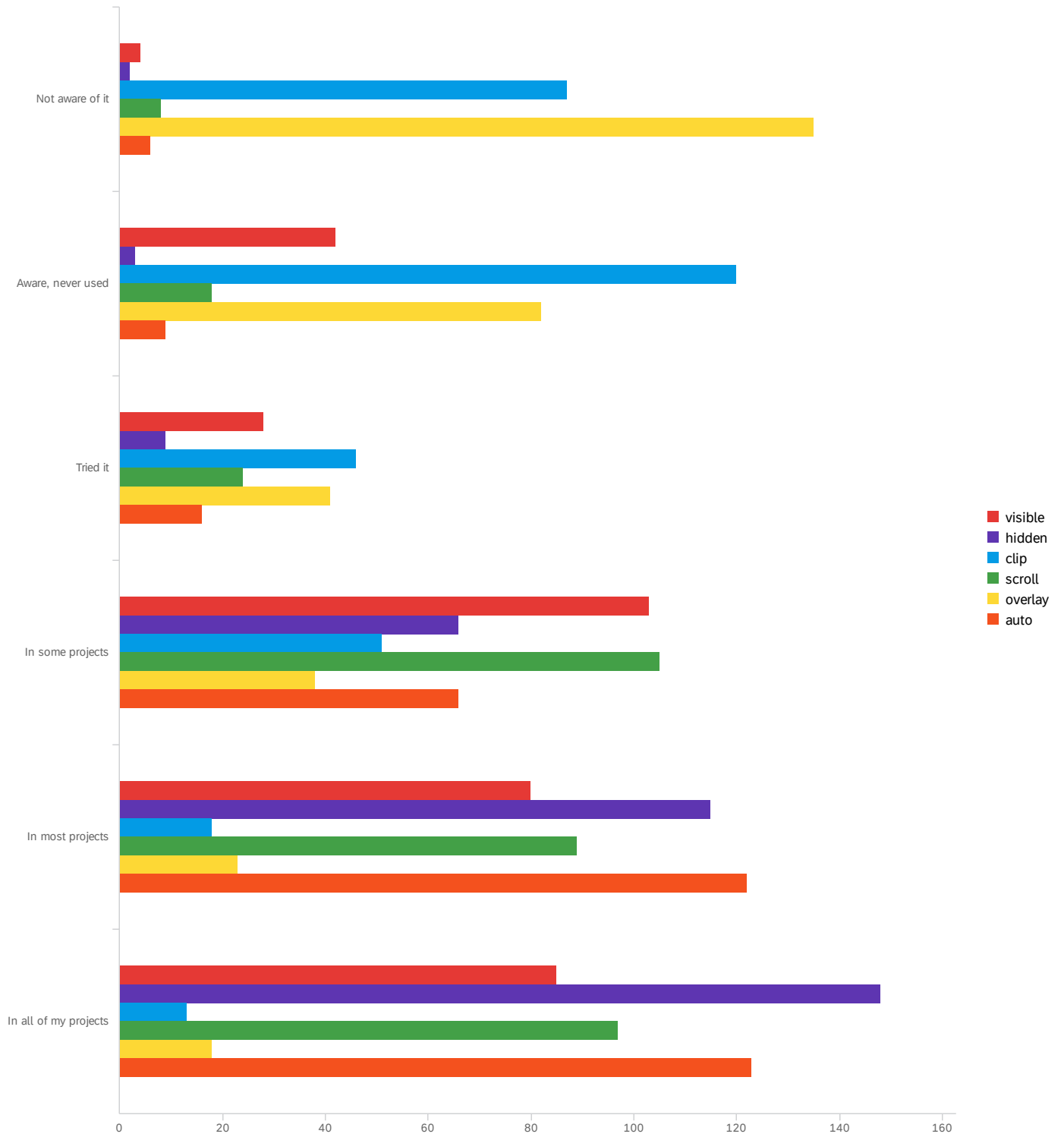
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	@scroll-timeline	1.00	6.00	1.43	0.85	0.72	343

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
2	scrollbar-color	1.00	6.00	3.44	1.90	3.59	343
3	scrollbar-width	1.00	6.00	3.51	1.98	3.93	345
4	scroll-padding & scroll-margin	1.00	6.00	2.93	1.96	3.83	344
5	scrollbar-gutter	1.00	6.00	2.34	1.80	3.26	345
6	scroll-snap-stop	1.00	6.00	2.62	1.80	3.24	345

#	Field	Not aware of it	Aware, never used	Tried it	In all of my projects	In most projects	In some projects	Total
1	@scroll-timeline	69.68% 239	23.03% 79	4.66% 16	0.87% 3	0.29% 1	1.46% 5	343
2	scrollbar-color	17.49% 60	23.03% 79	21.57% 74	2.04% 7	7.87% 27	27.99% 96	343
3	scrollbar-width	20.87% 72	20.29% 70	16.23% 56	3.48% 12	8.12% 28	31.01% 107	345
4	scroll-padding & scroll-margin	33.14% 114	21.51% 74	15.70% 54	0.87% 3	6.10% 21	22.67% 78	344
5	scrollbar-gutter	48.99% 169	21.16% 73	10.43% 36	0.87% 3	3.19% 11	15.36% 53	345
6	scroll-snap-stop	35.36% 122	26.67% 92	16.23% 56	1.16% 4	3.19% 11	17.39% 60	345

Showing rows 1 - 6 of 6

# Q17 - Have you used CSS overflow values?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	visible	1.00	6.00	4.37	1.33	1.77	342

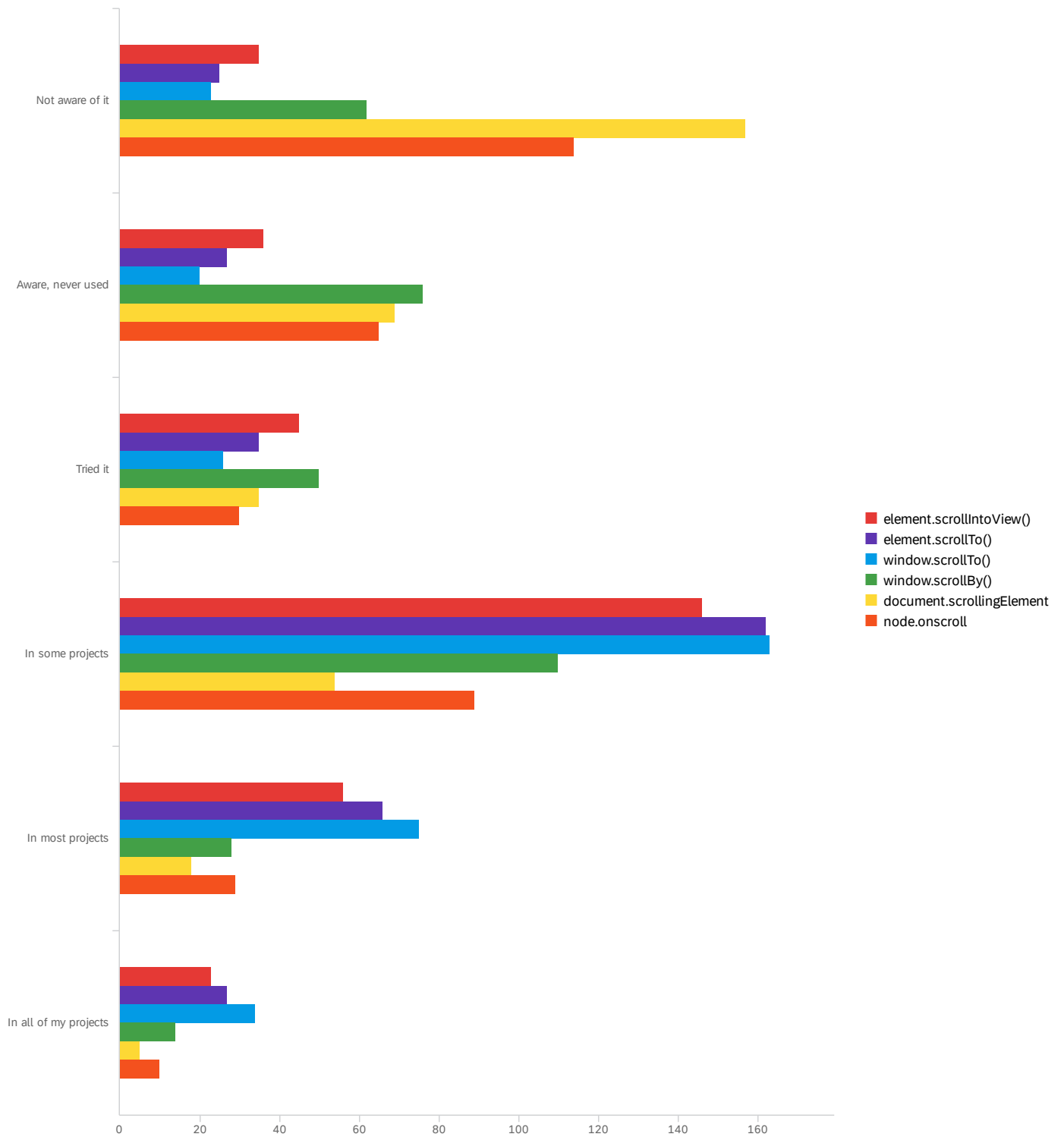


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
2	hidden	1.00	6.00	5.14	0.94	0.88	343
3	clip	1.00	6.00	2.50	1.36	1.86	335
4	scroll	1.00	6.00	4.58	1.25	1.55	341
5	overlay	1.00	6.00	2.36	1.52	2.31	337
6	auto	1.00	6.00	4.92	1.12	1.25	342

#	Field	Not aware of it	Aware, never used	Tried it	In some projects	In most projects	In all of my projects	Total
1	visible	1.17% 4	12.28% 42	8.19% 28	30.12% 103	23.39% 80	24.85% 85	342
2	hidden	0.58% 2	0.87% 3	2.62% 9	19.24% 66	33.53% 115	43.15% 148	343
3	clip	25.97% 87	35.82% 120	13.73% 46	15.22% 51	5.37% 18	3.88% 13	335
4	scroll	2.35% 8	5.28% 18	7.04% 24	30.79% 105	26.10% 89	28.45% 97	341
5	overlay	40.06% 135	24.33% 82	12.17% 41	11.28% 38	6.82% 23	5.34% 18	337
6	auto	1.75% 6	2.63% 9	4.68% 16	19.30% 66	35.67% 122	35.96% 123	342

Showing rows 1 - 6 of 6

## Q18 - Have you used JavaScript scroll properties?



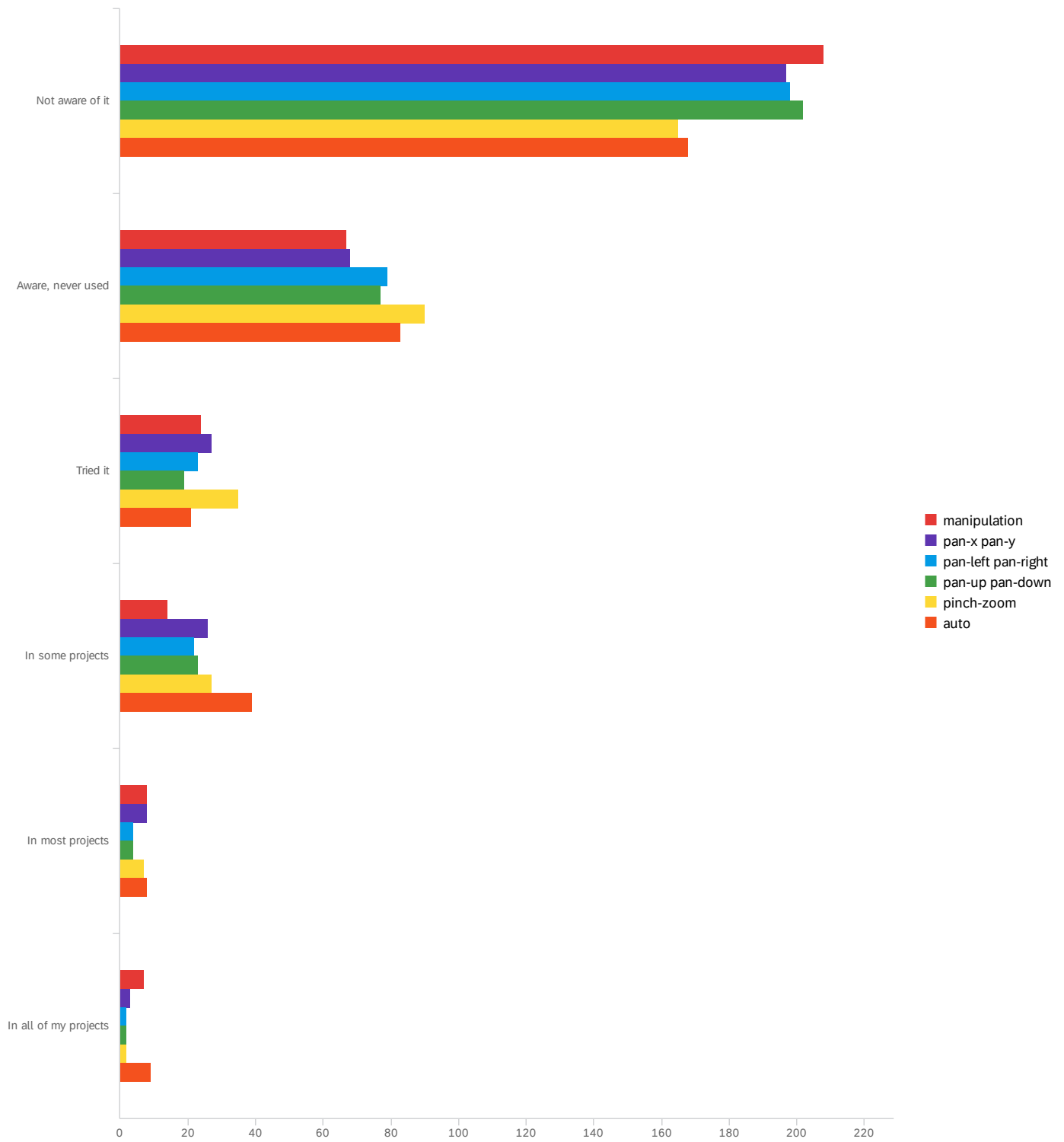
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	element.scrollIntoView()	1.00	6.00	3.65	1.34	1.79	341

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
2	element.scrollTo()	1.00	6.00	3.87	1.25	1.57	342
3	window.scrollTo()	1.00	6.00	4.02	1.24	1.54	341
4	window.scrollBy()	1.00	6.00	3.02	1.41	1.98	340
5	document.scrollingElement	1.00	6.00	2.18	1.38	1.89	338
6	node.onscroll	1.00	6.00	2.66	1.52	2.30	337

#	Field	Not aware of it		Aware, never used		Tried it		In some projects		In most projects		In all of my projects		Total
1	element.scrollIntoView()	10.26%	35	10.56%	36	13.20%	45	42.82%	146	16.42%	56	6.74%	23	341
2	element.scrollTo()	7.31%	25	7.89%	27	10.23%	35	47.37%	162	19.30%	66	7.89%	27	342
3	window.scrollTo()	6.74%	23	5.87%	20	7.62%	26	47.80%	163	21.99%	75	9.97%	34	341
4	window.scrollBy()	18.24%	62	22.35%	76	14.71%	50	32.35%	110	8.24%	28	4.12%	14	340
5	document.scrollingElement	46.45%	157	20.41%	69	10.36%	35	15.98%	54	5.33%	18	1.48%	5	338
6	node.onscroll	33.83%	114	19.29%	65	8.90%	30	26.41%	89	8.61%	29	2.97%	10	337

Showing rows 1 - 6 of 6

# Q19 - Have you used scroll-related touch-action CSS properties?



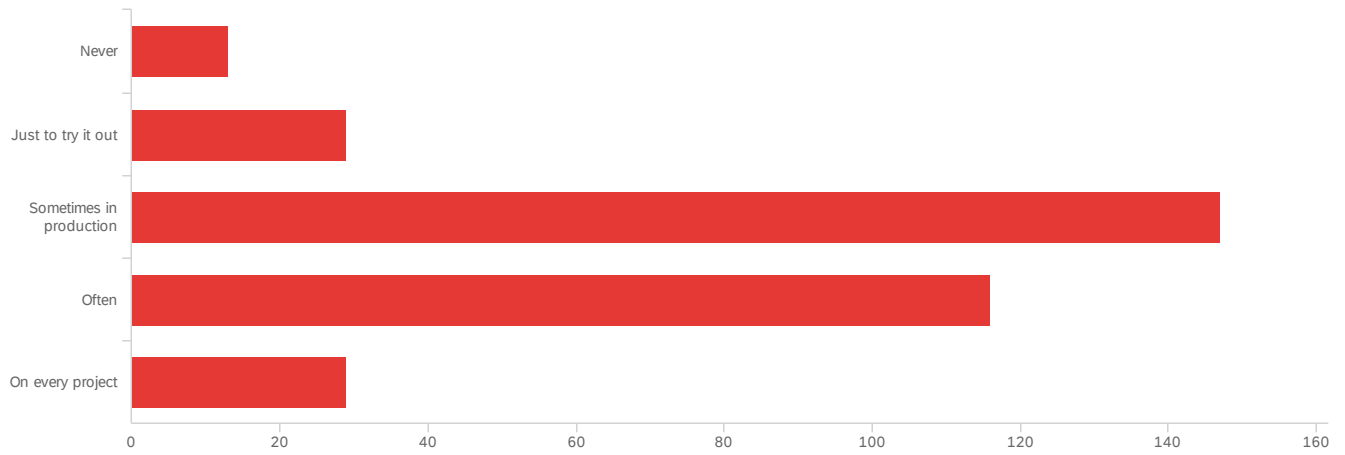
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	manipulation	1.00	6.00	1.68	1.16	1.34	328

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
2	pan-x pan-y	1.00	6.00	1.75	1.14	1.30	329
3	pan-left pan-right	1.00	6.00	1.66	1.02	1.03	328
4	pan-up pan-down	1.00	6.00	1.64	1.02	1.04	327
5	pinch-zoom	1.00	6.00	1.86	1.10	1.22	326
6	auto	1.00	6.00	1.97	1.31	1.71	328

#	Field	Not aware of it		Aware, never used		Tried it		In some projects		In most projects		In all of my projects		Total
1	manipulation	63.41%	208	20.43%	67	7.32%	24	4.27%	14	2.44%	8	2.13%	7	328
2	pan-x pan-y	59.88%	197	20.67%	68	8.21%	27	7.90%	26	2.43%	8	0.91%	3	329
3	pan-left pan-right	60.37%	198	24.09%	79	7.01%	23	6.71%	22	1.22%	4	0.61%	2	328
4	pan-up pan-down	61.77%	202	23.55%	77	5.81%	19	7.03%	23	1.22%	4	0.61%	2	327
5	pinch-zoom	50.61%	165	27.61%	90	10.74%	35	8.28%	27	2.15%	7	0.61%	2	326
6	auto	51.22%	168	25.30%	83	6.40%	21	11.89%	39	2.44%	8	2.74%	9	328

Showing rows 1 - 6 of 6

Q20 - Have you used inline (or horizontal) scrolling? For example: a carousel.

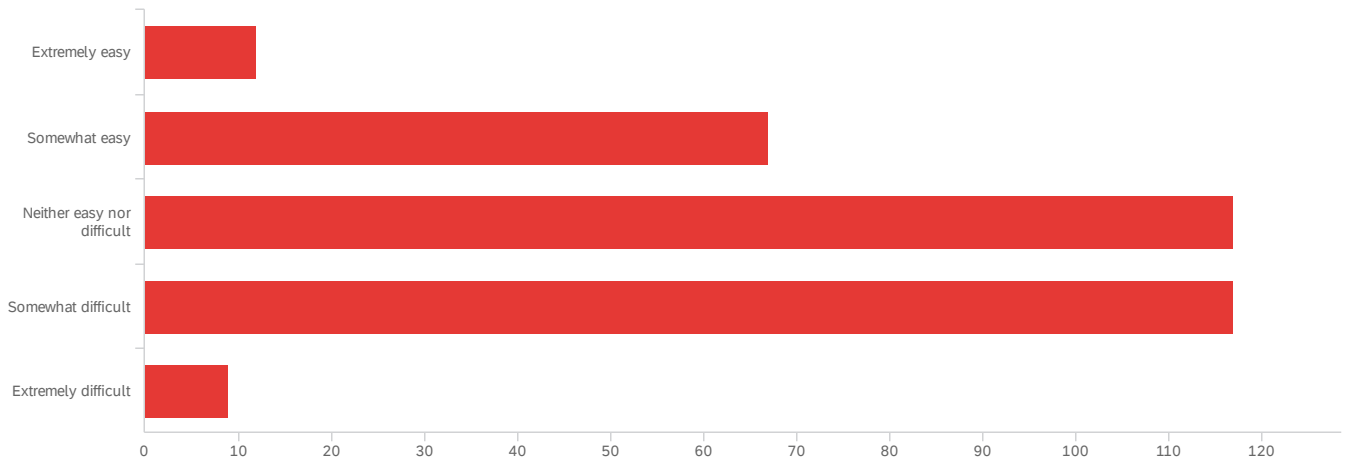


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Have you used inline (or horizontal) scrolling? For example: a carousel.	1.00	5.00	3.36	0.90	0.81	334

#	Field	Choice Count
1	Never	3.89% 13
2	Just to try it out	8.68% 29
3	Sometimes in production	44.01% 147
4	Often	34.73% 116
5	On every project	8.68% 29
		334

Showing rows 1 - 6 of 6

## Q20a - How difficult is it to manage inline (or horizontal) scrolling?

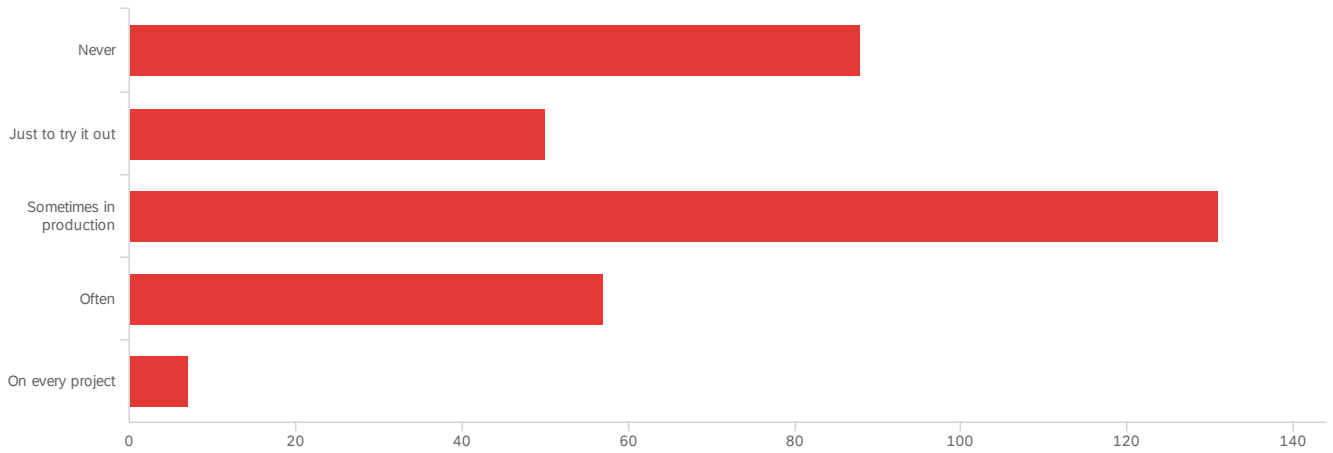


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How difficult is it to manage inline (or horizontal) scrolling?	1.00	5.00	3.14	0.90	0.81	322

#	Field	Choice Count
1	Extremely easy	3.73% 12
2	Somewhat easy	20.81% 67
3	Neither easy nor difficult	36.34% 117
4	Somewhat difficult	36.34% 117
5	Extremely difficult	2.80% 9
		322

Showing rows 1 - 6 of 6

Q21 - Have you used cyclical scrolling? For example: a carousel that cycles back to the first item.



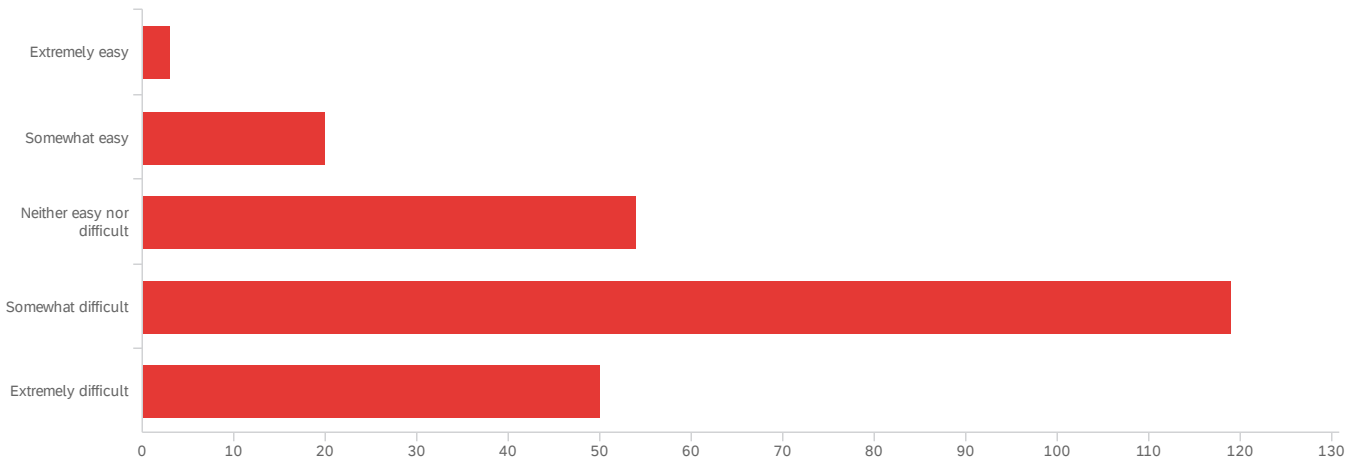
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Have you used cyclical scrolling? For example: a carousel that cycles back to the first item.	1.00	5.00	2.53	1.12	1.25	333

#	Field	Choice Count
1	Never	26.43% 88
2	Just to try it out	15.02% 50
3	Sometimes in production	39.34% 131
4	Often	17.12% 57
5	On every project	2.10% 7
		333

Showing rows 1 - 6 of 6



## Q21a - How difficult is it to manage cyclical scrolling?

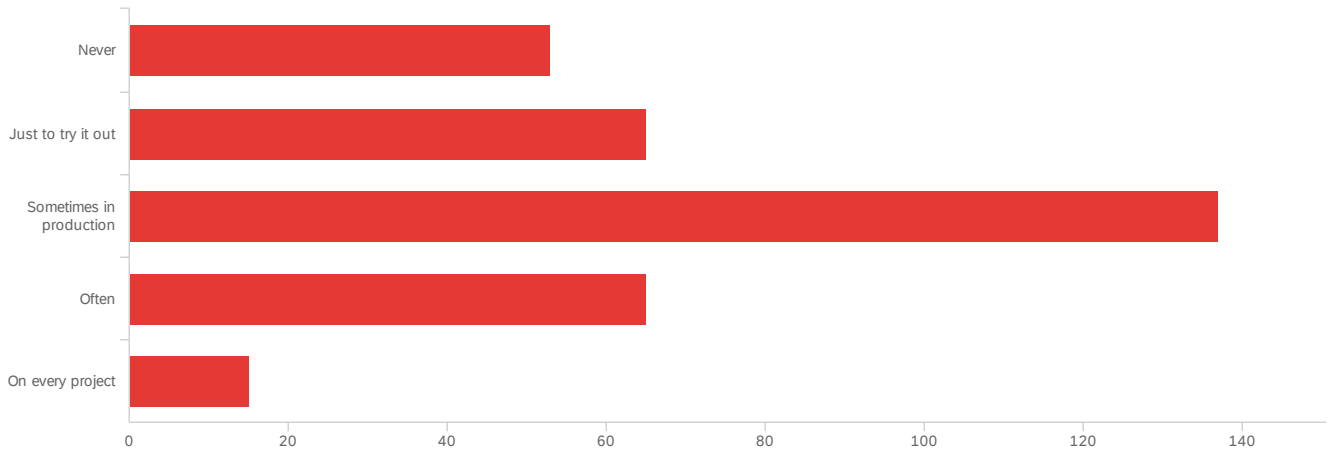


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How difficult is it to manage cyclical scrolling?	1.00	5.00	3.78	0.90	0.81	246

#	Field	Choice Count
1	Extremely easy	1.22% 3
2	Somewhat easy	8.13% 20
3	Neither easy nor difficult	21.95% 54
4	Somewhat difficult	48.37% 119
5	Extremely difficult	20.33% 50
		246

Showing rows 1 - 6 of 6

Q22 - Have you used infinite scrolling? For example: as scrolling reaches near the end of a page, additional items are displayed.

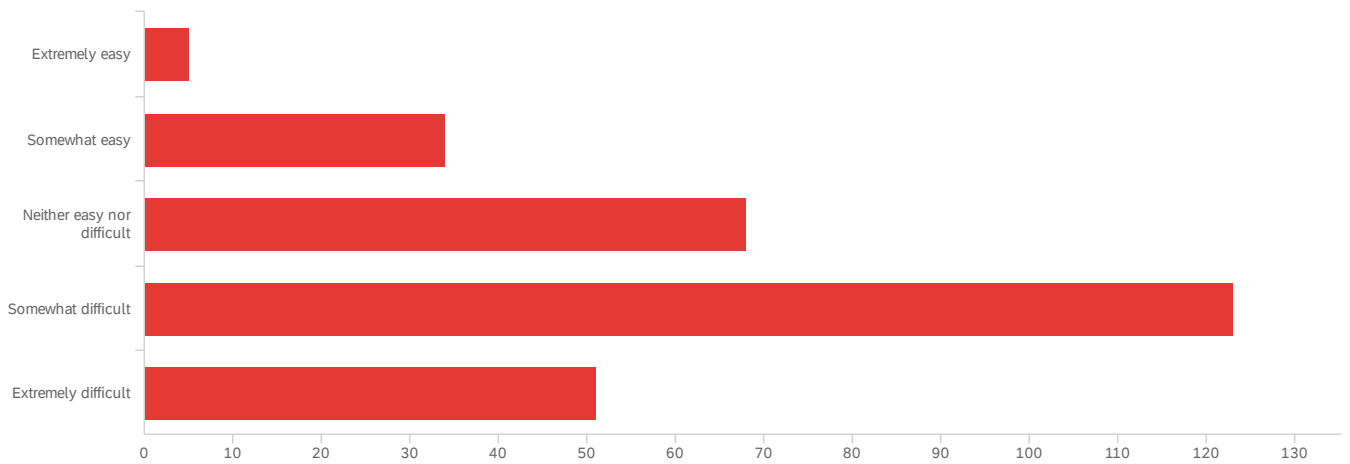


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Have you used infinite scrolling? For example: as scrolling reaches near the end of a page, additional items are displayed.	1.00	5.00	2.77	1.07	1.15	335

#	Field	Choice Count
1	Never	15.82% 53
2	Just to try it out	19.40% 65
3	Sometimes in production	40.90% 137
4	Often	19.40% 65
5	On every project	4.48% 15
		335

Showing rows 1 - 6 of 6

## Q22a - How difficult is it to manage infinite scrolling?

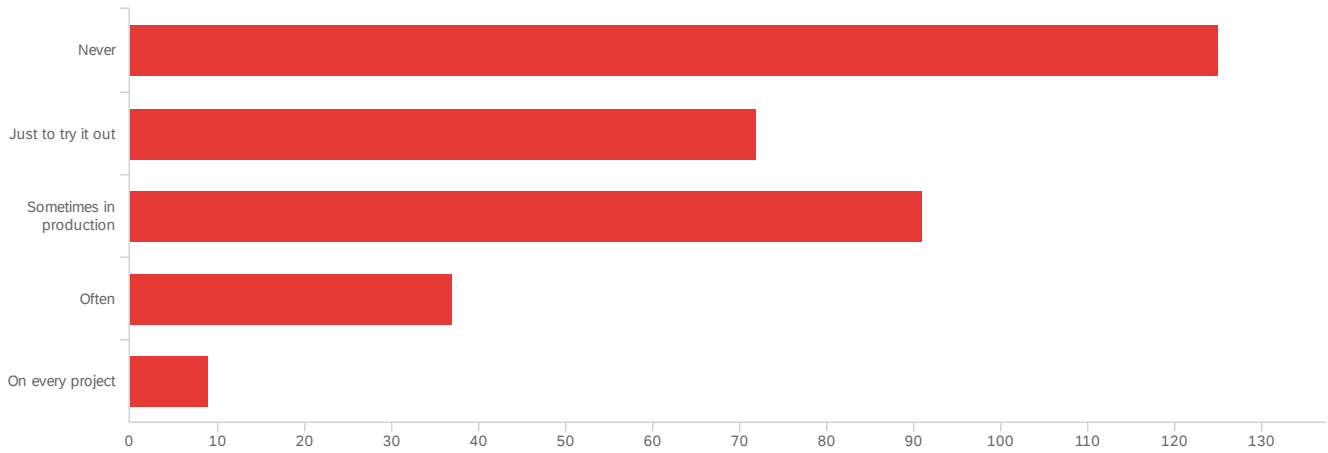


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How difficult is it to manage infinite scrolling?	1.00	5.00	3.64	0.97	0.94	281

#	Field	Choice Count
1	Extremely easy	1.78% 5
2	Somewhat easy	12.10% 34
3	Neither easy nor difficult	24.20% 68
4	Somewhat difficult	43.77% 123
5	Extremely difficult	18.15% 51
		281

Showing rows 1 - 6 of 6

Q23 - Have you used virtual scrolling? For example: items are appended or prepended to the scroll area depending on direction being scrolled.

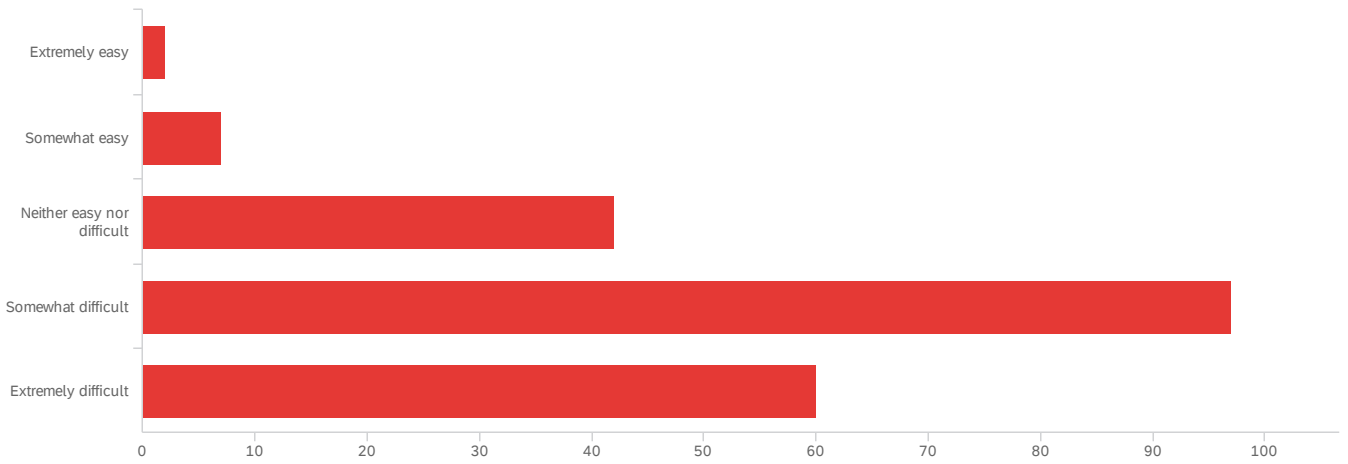


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Have you used virtual scrolling? For example: items are appended or prepended to the scroll area depending on direction being scrolled.	1.00	5.00	2.20	1.14	1.29	334

#	Field	Choice Count
1	Never	37.43% 125
2	Just to try it out	21.56% 72
3	Sometimes in production	27.25% 91
4	Often	11.08% 37
5	On every project	2.69% 9
		334

Showing rows 1 - 6 of 6

## Q23a - How difficult is it to manage virtual scrolling?



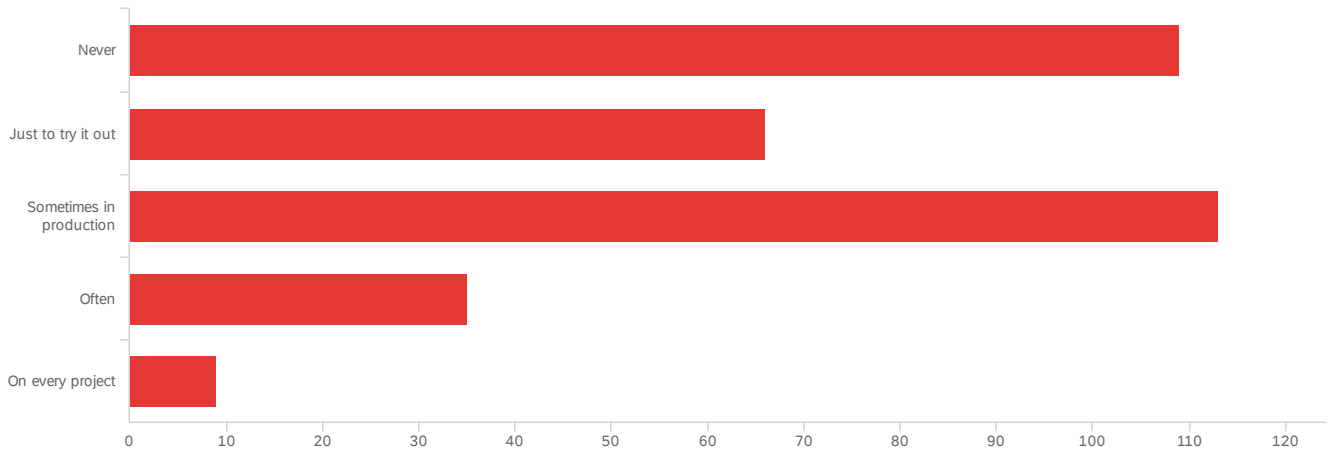
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How difficult is it to manage virtual scrolling?	1.00	5.00	3.99	0.84	0.71	208

#	Field	Choice Count
1	Extremely easy	0.96% 2
2	Somewhat easy	3.37% 7
3	Neither easy nor difficult	20.19% 42
4	Somewhat difficult	46.63% 97
5	Extremely difficult	28.85% 60
		208

Showing rows 1 - 6 of 6

## Q24 - Have you used scroll-linked, scroll-triggered interactions and animations? For

example: as a page area is scrolled, its position sets the value of a timeline.

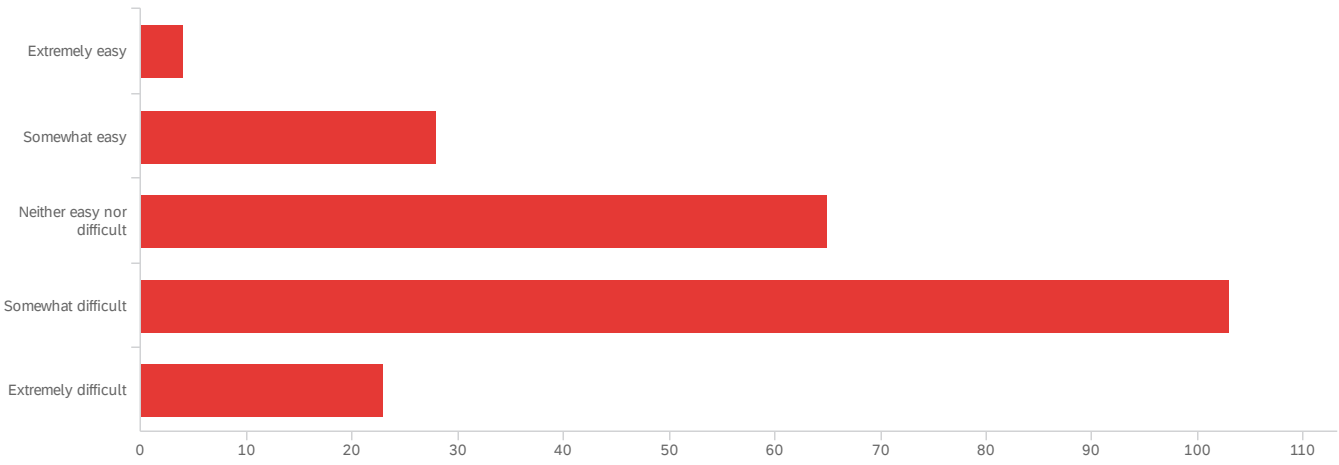


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Have you used scroll-linked, scroll-triggered interactions and animations? For example: as a page area is scrolled, its position sets the value of a timeline.	1.00	5.00	2.30	1.11	1.24	332

#	Field	Choice Count
1	Never	32.83% 109
2	Just to try it out	19.88% 66
3	Sometimes in production	34.04% 113
4	Often	10.54% 35
5	On every project	2.71% 9
		332

Showing rows 1 - 6 of 6

# Q24a - How difficult is it to manage scroll-linked, scroll-triggered interactions and animations?

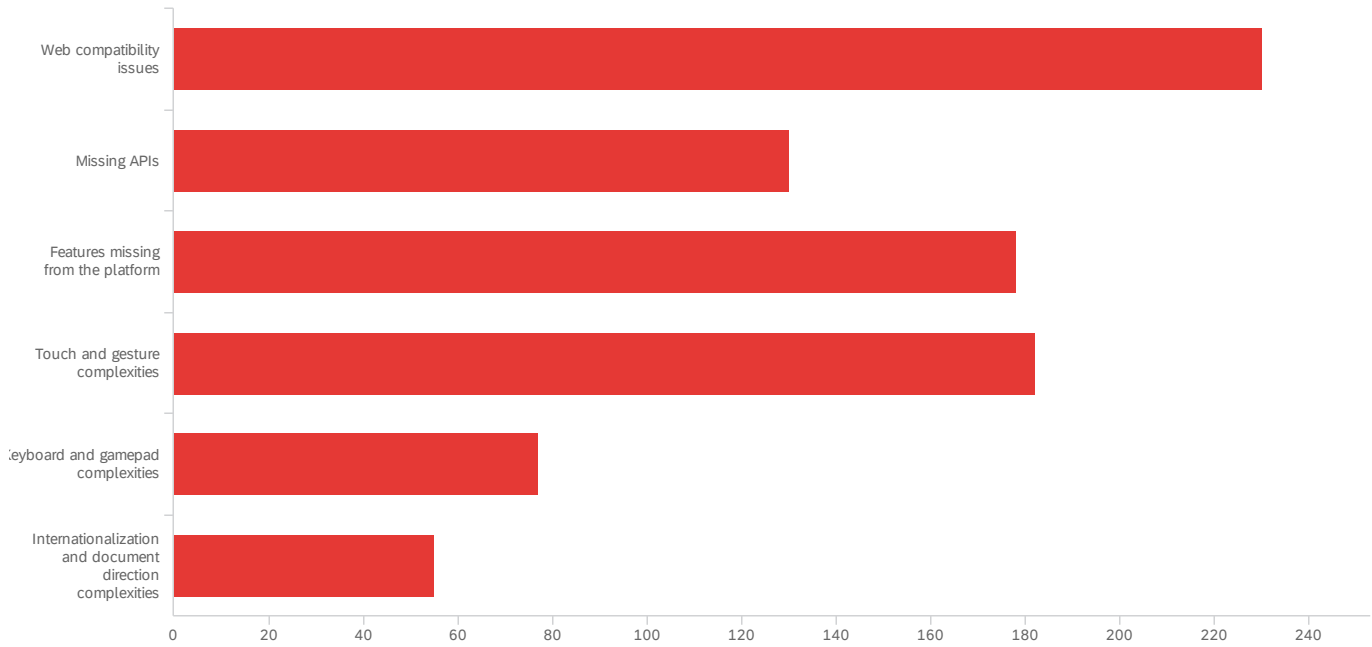


#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	How difficult is it to manage scroll-linked, scroll-triggered interactions and animations?	1.00	5.00	3.51	0.90	0.81	223

#	Field	Choice Count
1	Extremely easy	1.79% 4
2	Somewhat easy	12.56% 28
3	Neither easy nor difficult	29.15% 65
4	Somewhat difficult	46.19% 103
5	Extremely difficult	10.31% 23
		223

Showing rows 1 - 6 of 6

Q25 - Overall, have you experienced any of the following difficulties when implementing the scroll features described above?



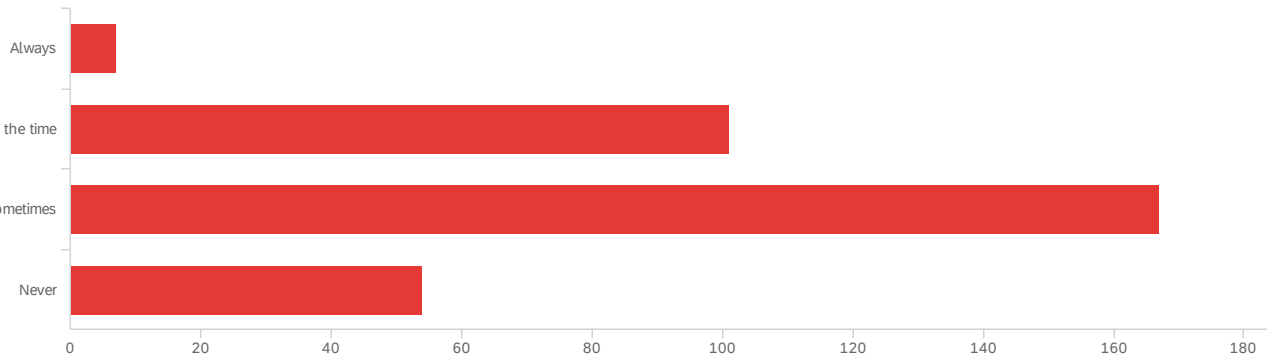
#	Field	Choice Count
1	Web compatibility issues	27.00% 230
2	Missing APIs	15.26% 130
3	Features missing from the platform	20.89% 178
4	Touch and gesture complexities	21.36% 182
5	Keyboard and gamepad complexities	9.04% 77
6	Internationalization and document direction complexities	6.46% 55

852

Showing rows 1 - 7 of 7



## Q26 - Can you build scroll interactions on websites and web apps that work as well as for native apps?



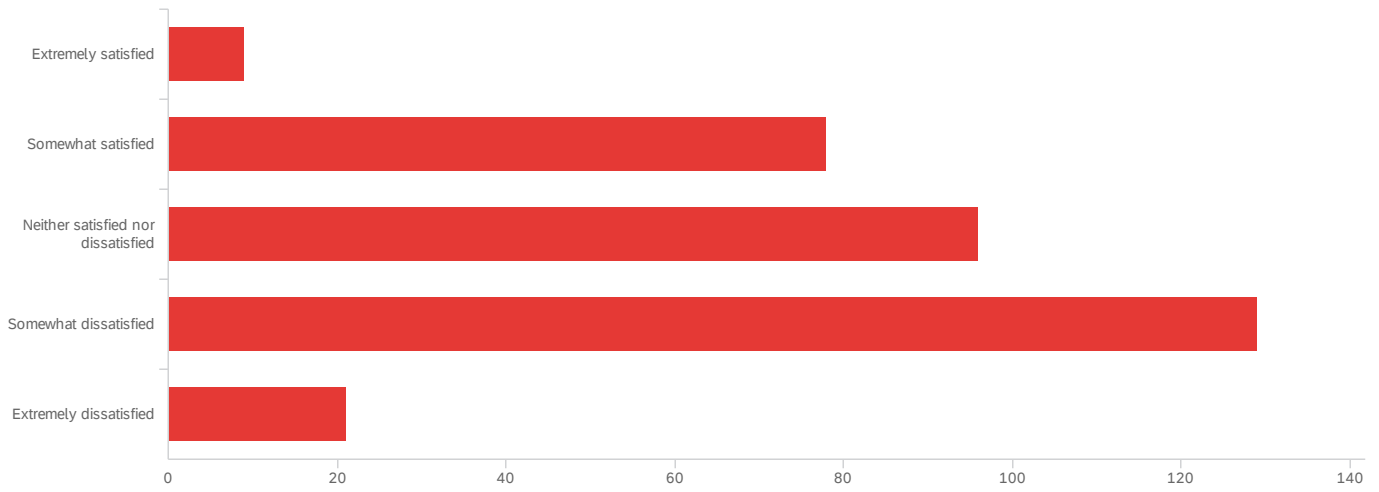
#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Can you build scroll interactions on websites and web apps that work as well as for native apps?	1.00	4.00	2.81	0.72	0.52	329

#	Field	Choice Count
1	Always	2.13% 7
2	Most of the time	30.70% 101
3	Sometimes	50.76% 167
4	Never	16.41% 54

329

Showing rows 1 - 5 of 5

## Q27 - Overall, how satisfied are you with building scroll interactions on the web?



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Overall, how satisfied are you with building scroll interactions on the web?	1.00	5.00	3.23	0.97	0.93	333

#	Field	Choice Count
1	Extremely satisfied	2.70% 9
2	Somewhat satisfied	23.42% 78
3	Neither satisfied nor dissatisfied	28.83% 96
4	Somewhat dissatisfied	38.74% 129
5	Extremely dissatisfied	6.31% 21
		333

Showing rows 1 - 6 of 6

## Q28 - Anything else you'd like to share about scrollable interactions?

Anything else you'd like to share about scrollable interactions?

It would be nice to control in which direction the visible content is shifting if the height of a (horizontal overflowing) container changes. Example: testimonial carousel with 5 slides. Each slide is one review. If review 2 is visible and I click a button to see review 3 (which has more content) then this extra content gives the testimonial section more height. Sometimes the following content gets pushed away. Sometimes the content before the testimonial section moves up so that the bottom border of the testimonial container does not move. But the top is moving up. Very strange.

standardize stuff like `"::-webkit-scrollbar"`, `"::-webkit-scrollbar-thumb"`, ...

In a container with `overflow: scroll;`, the content does not stick out, but I need a way to make the content stick out with `position: absolute;`. For example, if I have a tooltip inside a scrolling container, I may want to have only the tooltip protrude outward. Another thing to mention is that when using virtual scrolling, it is difficult to restore the scrolling position after a page transition.

Libraries end up filling so many of the missing scroll/scroll animation gaps in the platform that they are the defacto method used by so many devs and agencies, it's just much more efficient. I wish this wasn't the case and we had some native web APIs to use as more and more websites are expected to "come to life". More libraries = bigger bundles. Less native APIs = less FPS ;)

Thank you for thinking and processing scrolling in a purposeful meaningful way. I'm excited.

loving most of this scroll features and media queries, can't wait to be able to use them in production as it gets support across all browser, hope it doesn't take too long cause this makes everything so much better and fun.

Scroll is painful(

For me, the biggest gap in APIs is where CSS doesn't provide a way to declare functionality that could be static, requiring JS to accomplish those tasks. Example: scroll-snap is great for implementing a basic carousel, but I've found it challenging to display a dot nav which highlights the dot corresponding to the item currently scrolled into view. I recently accomplished this with `position: sticky;` but it's far from obvious and required a fair bit of hackery. Another example: providing a "scroll shadow" is a helpful visual hint that there is overflowing content. But getting these to work across browsers can be a challenge. You can see on this blog post (<https://eyelidlessness.github.io/blog/2021/03/what-the-art-part3-implementation/>), mobile Safari will hide the shadow as expected when scrolling to the end of the points picker widgets, but not for the code blocks. The apparent difference is that the former is a flex container, the latter a grid. It looks like @scroll-timeline will help significantly as it becomes available. But there are other opportunities to improve pure HTML/CSS static use cases: - It would be great to have native semantic HTML to initiate relative or arbitrary distance scrolling like `page-[direction]` or similar. - It would be great to be able to alter positioning behavior of children with CSS, for instance directly specifying scroll speed (parallax etc) and inverting scroll direction (both of these would have helped my carousel dot highlight use case). Minor UX note: I'm replying to this on mobile and this textarea allows a surprisingly large amount of feedback, but it only displays three lines of text. Which itself has presented scrolling challenges ☹️

Styling scrollbars is much easier but still has some quirks - custom styled ones on desktop browser seem to have ui delay.

The less customization the better. Rely on browser standard functionality for the best accessibility.

For me, scrollbars are probably the most frustrating elements to work with, when creating a web app.

There should be more focus on when `*not*` to play scroll animations when combined with ``prefers-reduced-motion``.

Many things have changed in a positive direction within the last two years (intersection observer, scroll snapping, overflow behavior). But honestly the most pressing pain points for me are: - lack of ability to easily detect scroll end (including "bouncing") - inconsistent handling of padding of the scroll container across browsers - threshold of interaction observer is not precise (eg. setting `threshold=1` is not met most of the time when scroll container and observed item have same size) - lack of ability to detect support for `ScrollIntoViewOptions` like `"block: center"`

Shadows ! Pure CSS Scroll shadows are so hard to do well with current APIs. A scroll-shadow property would be awesome

Anything else you'd like to share about scrollable interactions?

On of my biggest issues with scrolling is performance. It's very difficult to animate things in while scrolling without dropping any frames.

Because of my latest experience, I feel like the having huge lists of elements always requires a custom JS based solution, being trivial on native mobile platforms. I just discovered that touch interactions can be customized with CSS, I am going to try that.

Windows Edge has native smooth scrolling. If every browser had this feature it'd greatly improve the challenge of coding a custom solution that is accessible and performant.

Scroll speed that is induced by scroll snap is too slow.

One of the very difficult part are scrollable area that aren't always scrollable, but may become scrollable as the content AND/OR the viewport changes.

Native quality scroll effects are still not quite there, especially on iOS. Scroll regions lock up, scroll events aren't synced to frame with the push to passive listeners which results in a poor animation and finger following. Properties that often need to be animated during scroll bouncing are not efficient to animate, such as height (for example a profile page where the top grows during an over scroll).

Position a floating button with position :fixed; bottom: 2em; Then, using Chrome on Android, scroll up/down quickly (momentum scrolling) so that the url bar is shown/hidden, and then stop the scrolling by tapping the screen. Now, the tap target of the button is in the wrong position, often causing the user to tap on interactive elements in the background. Really really annoying

I need scrollIntoView({behavior: 'not-smooth'}) to cancel out when the root element has scroll-behavior: smooth and I want to snap to an element. 'auto' doesn't do that and thats the only option other than smooth.

My main issues are with controlling the layout and position of scroll bars, and how they affect other elements. When I last had major issues with this, scroll gutter was the proposed solution, but wasn't well supported. I'm not sure the current situation.

Recycler Views and Infinite Scrolling would be amazing to have better solutions built-in to the web

Most of the problems I encounter stem from only being able to have one smooth scrolling element on a page at a time

High speed vertical scroll where the scroll snaps to a position when the user inadvertently touches the scroll bar is an INFURIATING user experience. There is a lack of native behaviours for horizontal scrolling on mobile devices results in a need for third party dependencies and or large portions of custom coding that often negatively impacts core vitals.

I've used ScrollTrigger. It's great but the performance gets worse as it's not native and I would like to not need an external library.

why is safari doing everything differently regarding scroll ? overflow hidden does not work on body, smooth behavior missing....

Most of the time I use libraries that wrap user interaction for different browser/devices.

One feature that I would really love to be able to accomplish with native scrolling is to allow a sort of overflow:auto-visible, where content is scrollable but does not clip outside of the box's bounds. This could be used to accomplish MacOS style "frosted glass" translucency for content that scrolls outside of one container, and "under" another. I know of no way to accomplish this today without resorting to custom scrolling, since there is no way to control overflow like I mentioned above. Another alternative may be to allow arbitrary positioning, or at least some control, over scrollbar position. So you could give, say, 100px of top margin to the scrollbar then overlay the "frosted glass" over that top 100px of the scrollable container.

wanted to animate elements when box is in viewport. Had problems because the start-position was translateY(-xx) and anchors etc broke because of that.

do not enforce fancy but bad UX just to satisfy designers, we need no APIs for that and no code for that, just leave things as they are and it works best

Anything else you'd like to share about scrollable interactions?

Documentation of features, and the naming of the features themselves are very hard to access and unintuitive. The Gif examples used in this survey should be \*required\* on any documentation page for visitors to understand what they're dealing with.

From modern browsers safari has strong differences in the end user experience when doing animations on scroll.

Maybe web.dev is worth having more articles about scrolling capabilities

I wish there was a native horizontal scroll solution!

Smooth scroll, scroll snap, snap for sliders, and over scroll are huge for my work. Thanks!

I hate these as a user. I hate that developers have more and more ways to make custom interactions, making the web less and less consistent. I'll often hit the back button if a site uses custom scrollable interactions.

On sites I've worked on, I feel like scrolling has just been accepted as whatever your OS displays it as. Making custom scrolling options native to all browsers and having one library for it all is key. I don't feel like scrolling styles should be as hard to manage as they are. Thanks for the survey!

iOS/iPadOS compatibility is the worst

I've seen a lot of new properties come out but the accessibility and cross compatibility is lacking and that's a big no when it comes to real projects

Improved scroll-snap support cross browser support would be a game changer for the kinds of things I do as a developer.

In flexbox and grid, browsers are inconsistent about whether a scrollbar should subtract from content or add. Asynchronous scrolling presents problems if you need to synchronize (e.g. more complicated sticky positioning done in JS) but there's no way to turn it off.

Everything to do with cyclical carousels feels like a hack, need to be pretty smart to be able to implement them.

Thanks for make us the web dev easier!

I often use plugins like GSAP for scroll animations and Flickity for carousels which make my life much easier.

1. There are still a lot of bugs around scrolling in 2021...: - no way to prevent body scroll in all situation on iOS - scroll-snap-stop is buggy in all browsers implementing it - scroll chaining doesn't work horizontally on iOS in some situations - overscroll-behaviour: none on an element causes body scroll to be blocked in Chromium - clip-path in combination with scrolling is buggy in WebKit - scroll-snapping is buggy in some situations in Firefox - ... 2. There are many unspecified aspects about scrolling which causes the experience to be uneven between browsers, but also OSs and devices, and prevent relying on them. Like: - what should happen with the browser UI when scrolling either the body or an element - what should happen when there is only one snap-point defined in a scroll-snapping context? - when is scroll chaining triggered in case of overscroll? - what should happen scrolling context which allows scrolling in both direction, when starting to scroll in one direction, and then going in the other? Should it be totally fluid, or should the browser weirdly keep scrolling in the same direction? - ... 3. We need more control over the current behaviour of scrolling: - we should have control over the easing function and duration of scroll-snapping (with restrictions) - we should be able to define initial scroll position in CSS, necessary for some things in SSR - we should have more control over the easing and duration of smooth-scrolling, and it should run at 60fps in all browsers - we should have more control over overshooting when overscrolling in WebKit, and this should be spec'ed and adopted by other browsers - overscroll-behaviour should be adopted widely - scroll-linked animations should be adopted widely :) - a more generic input-based animation spec should be developed - ... Happy to discuss this further: @brunostasse on Twitter

Still miss features for simple smooth scrolling, forcing me to use virtual scroll libraries like locomotive or asscroll. This brings extra problems when trying to animate elements on scroll value, as it's takes time to calculate the new value. A smooth scrolling API would release the need for extras libraries, and much more robust usage of smooth and reactive animation. Many clients all for smooth scrolling, as it gives the visitor a sense of performance and elegance.

The biggest problem with most scroll-linked work on the web is that it's usually a bad idea. More useful than additional features or standards would be better product managers or stakeholders. Alas...

Anything else you'd like to share about scrollable interactions?

An overflow value that allowed content to overflow while still being scrollable would be really useful. A client wanted a centered carousel on their site where the items could display full width of the browser rather than clipping to the content width. I ended up having to use transforms to create the effect they wanted but it would have worked exactly as intended out of the box if I could apply 'overflow-x: visible-scroll' and get an 'overflow: visible' container with a scrollbar to scroll. It would also be useful if there was an option to swap the scrollbar for dots (or add dots as well) that can be styled as needed and provide the carousel experience by default. It's such a common UI pattern that shouldn't need to be done manually every time.

Recent features like scroll-snap or position: sticky really helped with development, only issue that often needs a workaround is position sticky table headers on a table that is scrollable horizontally without a contained height

Please keep it simple and stupid.

Animation in a world of async scrolling is pain 🤢 And address bars folding on scroll, pain 🤢

I usually use JS libraries for all of this

Make it easy to interact and same behavior in every platform

If I could make an arbitrary dictation to all web UX, I'd ban infinite scrolling \_that doesn't save its position in the URL\_.

Chrome, please fix scroll-snap

Infinite scrolls sucks; stop using it.

I have faced some tough time when trying to apply specific behaviors for: - snapping some children elements - scroll snap step-by-step (one gesture would scroll many steps at once instead of a single step) - circle scrolls - scroll behaviour (scroll on a fixed or absolute child that would not scroll the page itself) - hiding the scrollbar but keeping the scroll features working (on multiple platforms)

So unfortunate that <virtual-scroller> is no longer being pursued: <https://www.chromestatus.com/feature/5673195159945216> I was really looking forward to this.

Some serious work on scrollable interaction is very much needed. I especially think about smooth-scrolling, parallax, scroll-linked animations. Some improvement could be done with scroll-snap, like a way to indicate the state using dots.

There are so many bugs and inconsistencies between browsers for basic layout of scrollbars, their content and containers. Some are listed here: <https://github.com/chearon/scrollbugs>. It's virtually impossible to implement lock-step scrolling and other scroll-linked positioning effects now. We need these for actual application functionality, like implementing a spreadsheet with frozen panes (several table elements that scroll in sync) and big content areas with virtual scrolling.

Google Calendar is unusable with a Logitech MX mouse, which has a low-friction (i.e. infinite) scroll wheel. Imperceptible rotation of the wheel causes the calendar to fly forward/back months at a time making it nearly impossible to use. Google Maps does not suffer this issue with zooming, so it is solvable. To be able to use Google Calendar at all I have to use this extension, which an amazing saint has maintained through generations of it being broken by updates: <https://chrome.google.com/webstore/detail/google-calendar-scroll-di/nghndfiaocgpmcbeafghknkifgddebe?hl=en-US> I would love if Google would fix Google Calendar scrolling, or provide a config option to "disable scroll wheel" entirely.

I am not very experienced in using scrolling interactions.

Please add native Spatial Navigation with Shadow DOM (open) support.

Scrolling doesn't match requestAnimationFrame which makes scroll-animation async to native html scrolling.

Ease out Scroll Functions and Customisation for Scroll related interactions in near future and create awareness about the same

Anything else you'd like to share about scrollable interactions?

Even to this day, most applications of scrolling in/from/involving JavaScript feel sluggish across, especially on lower-end hardware, and it greatly limits the options I prefer to use. Anything using JS is only used as a very last resort, even though a lot of stuff would be so much easier with it.

I hope this is relevant. But scrolling areas with child elements that are also clickable / contain images can be extremely difficult on the web. You can end up dragging images and firing links while trying to scroll.

Debugging pinch-to-zoom and related multi-tap gestures is difficult, since there's no way to emulate multitouch interactions in DevTools. Being able to record and play back multitouch interactions (by recording one tap at a time) would be great.

I hate infinite scrolling for all usages other than brainless scrolling of random content. I also hate, as a developer, to have to scroll through devtools console only because console.groups don't get filtered (chromium ticket 1151616) I find it difficult to develop "enforced scroll" - not sure what is the jargon name - when you drag an element within an overflow container and you hold the element at the boundaries to force the container to scroll a bit and reveal the drop target.

Thank you for working on improvements in this area!

As glad as I am that chromium is supportive of advanced features especially scrolling as our daily consumption of content ie. blog posts and documentation is mostly long-form content, other browsers are missing key features and that is what makes scrolling a problem. Also, I wish there was a dedicated scrolling testing embedded into chrome dev tools as right now it is just an iterative process that result's in a good scrolling experience, and most often first versions don't work as intended. The fact that I was able to fill out a survey related to scrolling is an indicator of progress in a good direction, but it takes way too long to make the scrolling experience nice cross-platform/browser.

Having to account for padding on the end in horizontallay scrollable flex containers

It would be nice to have some kind of automatic scroll restoration API

Fixed elements (e.g. a sticky footer) being obscured by a soft keyboard is the biggest pain point I've faced recently (looking at iOS...). Caused a bug on iPad 12.1 that resulted in a significant drop in conversion.

The inconsistent styling of scrollbars is a good topic to ask about

Scrolljacking should be considered a crime.

Scroll events are difficult to build every browsers. Some browser is too slow, some browser does not support, sometimes using the Wheel event, sometimes using the scroll event.

**End of Report**