

# weird browsers

nlhtml5 — october 15th 2015







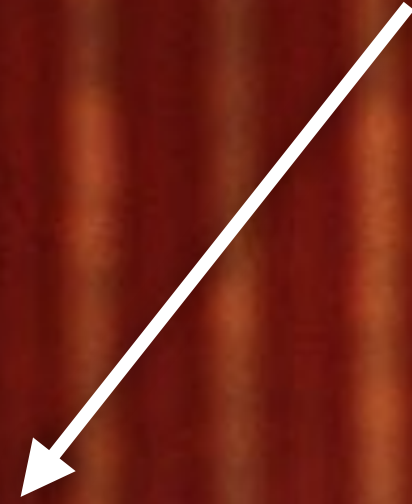




this slide is  
inspirational as *fuck*



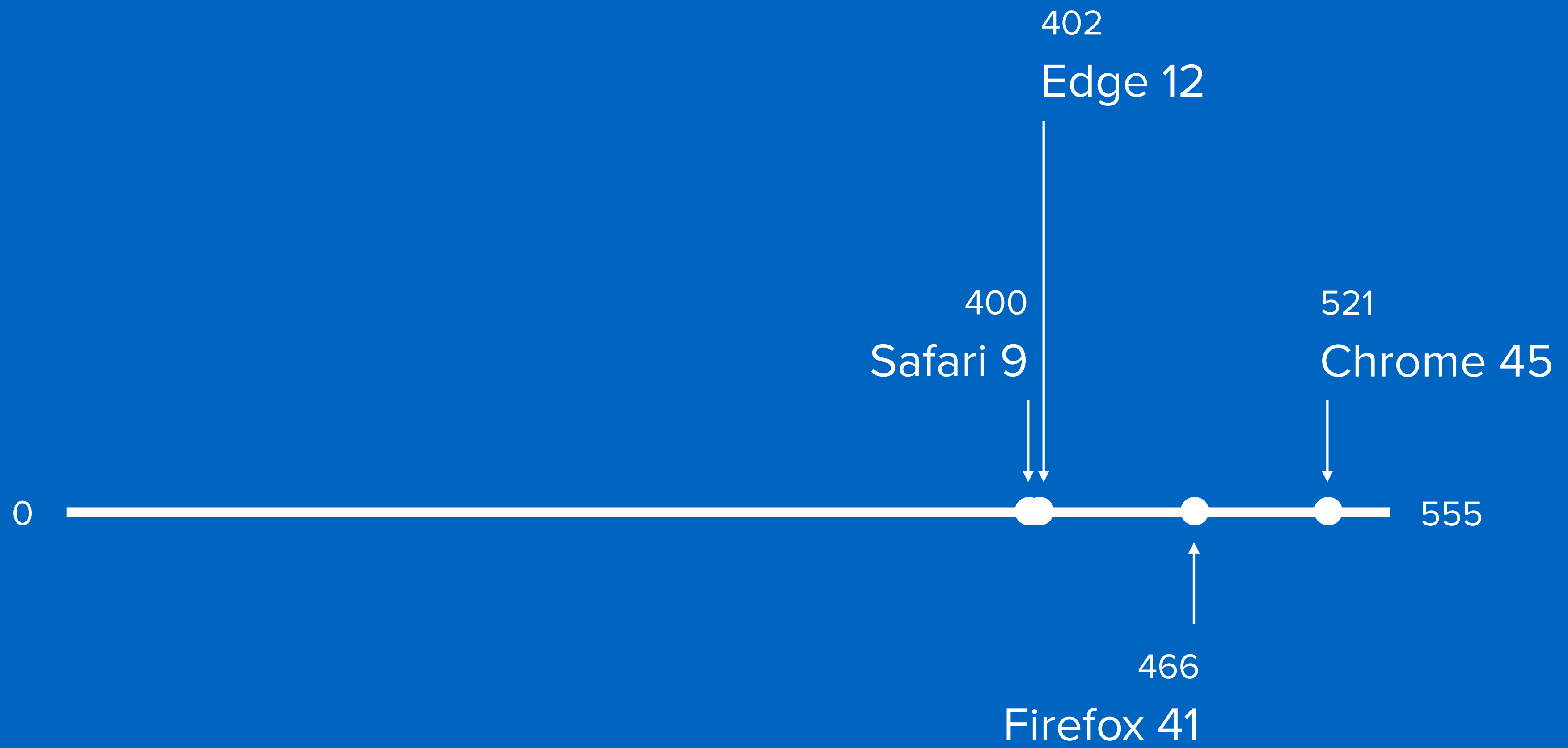
browser compatibility tester





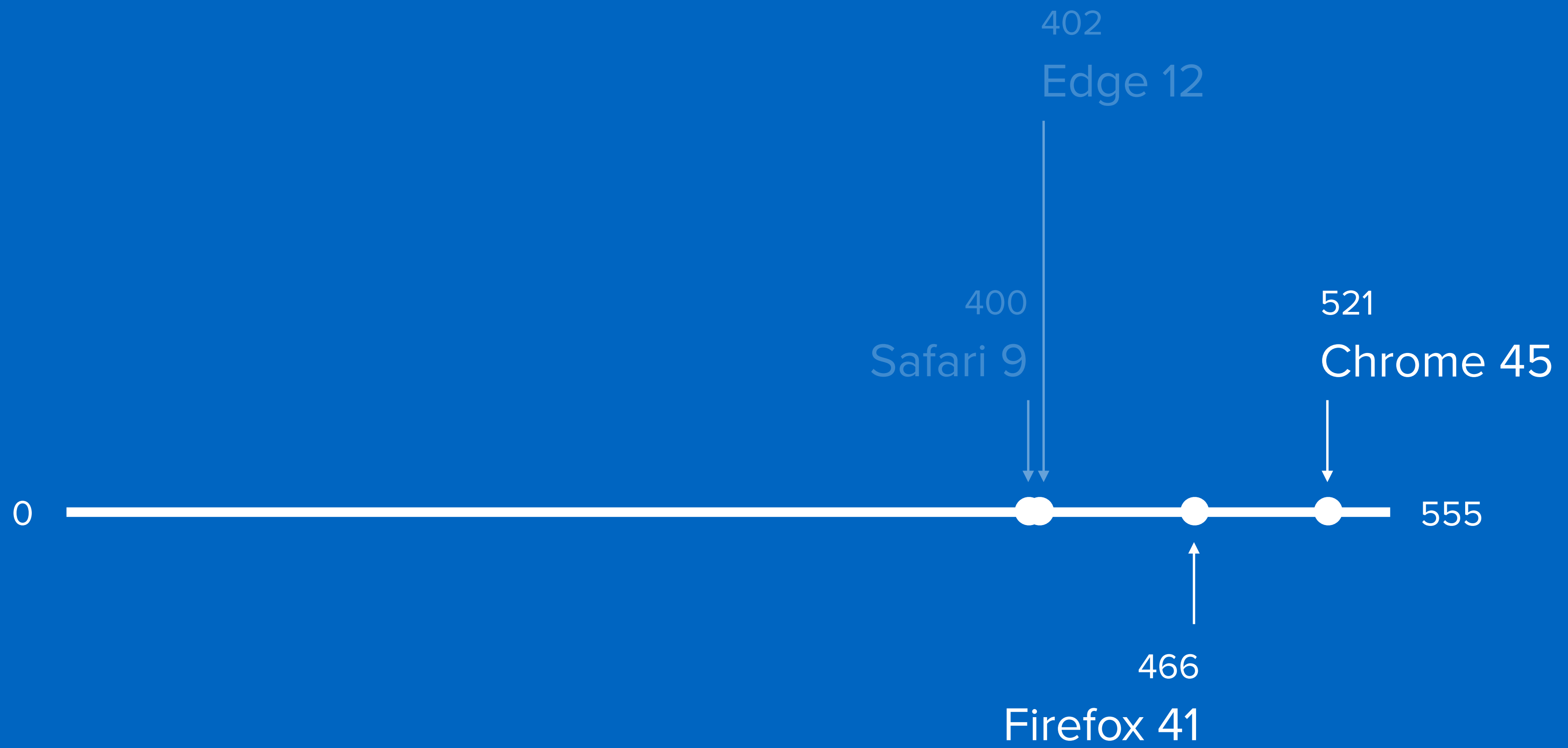






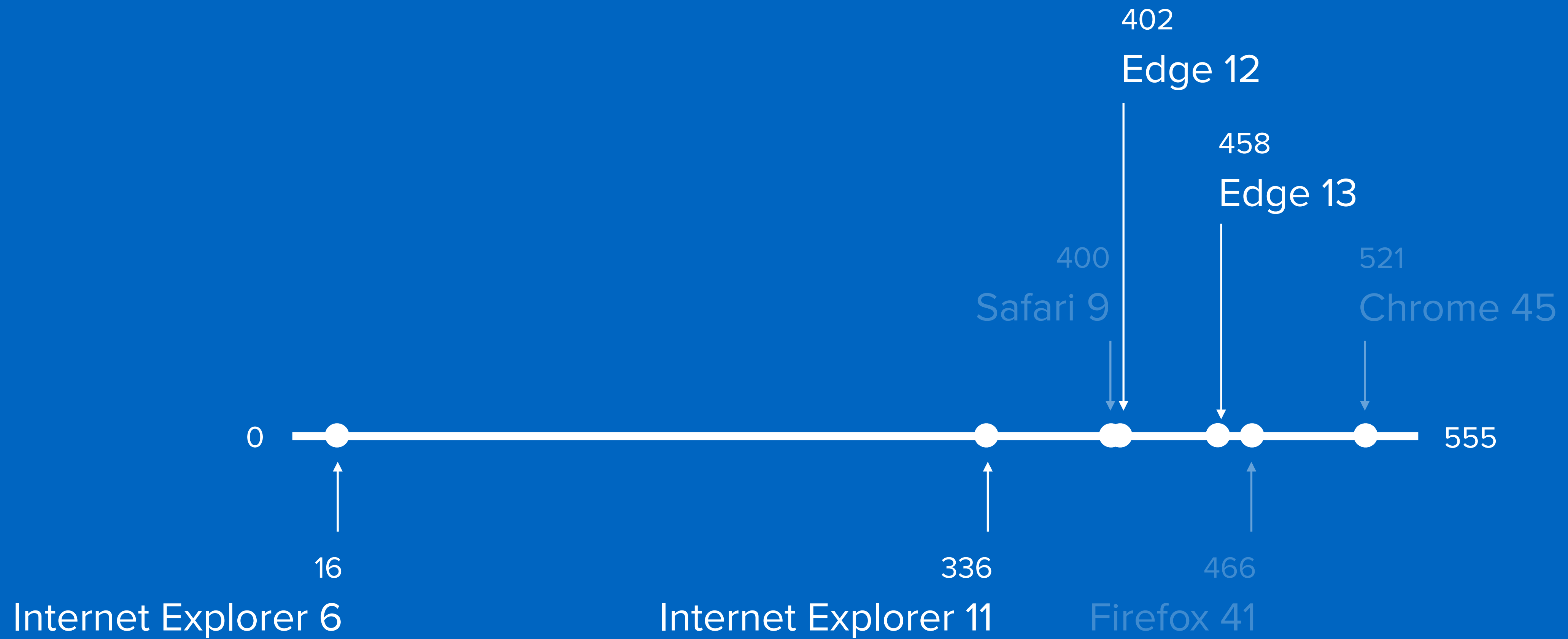
desktop browsers results on [html5test.com](http://html5test.com)





desktop browsers results on [html5test.com](http://html5test.com)





desktop browsers results on [html5test.com](http://html5test.com)







# weird browsers

nlhtml5 — october 15th 2015



**browsers and devices that do not  
adhere to current expectations**



```
if ('ontouchstart' in window) {  
    element.addEventListener('touchstart', function(e) {  
        ...  
    });  
}  
else {  
    element.addEventListener('click', function(e) {  
        ...  
    });  
}
```







weird browsers

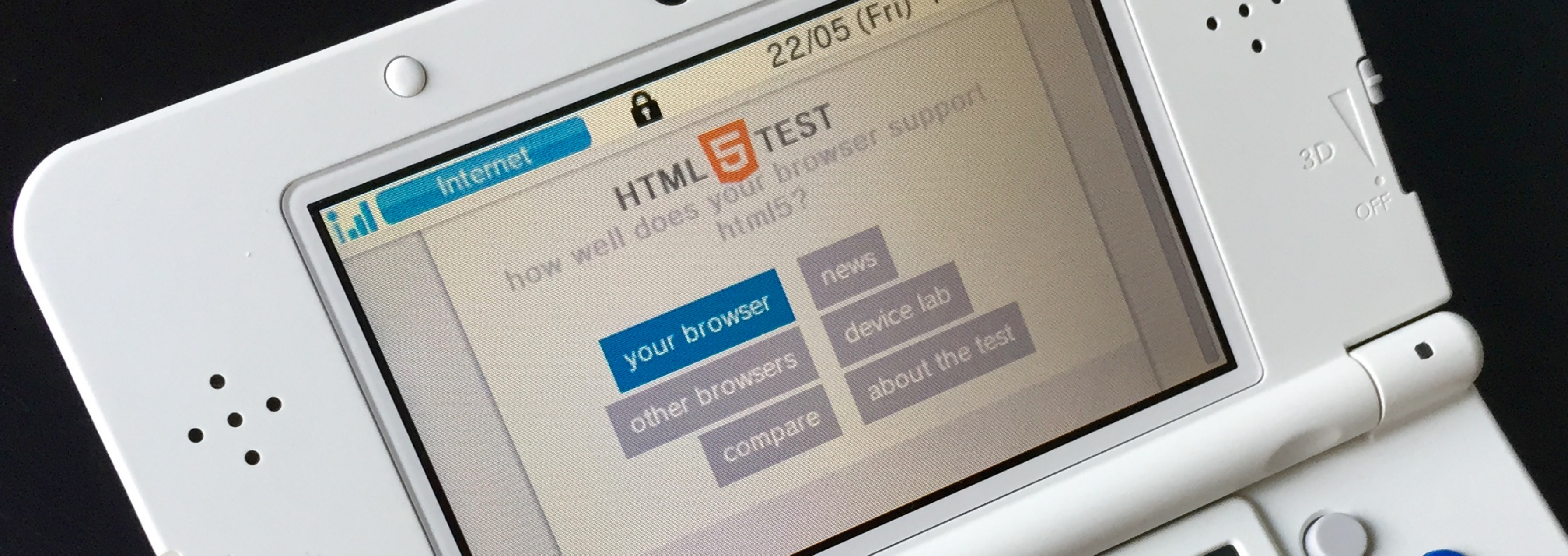
weird browsers?



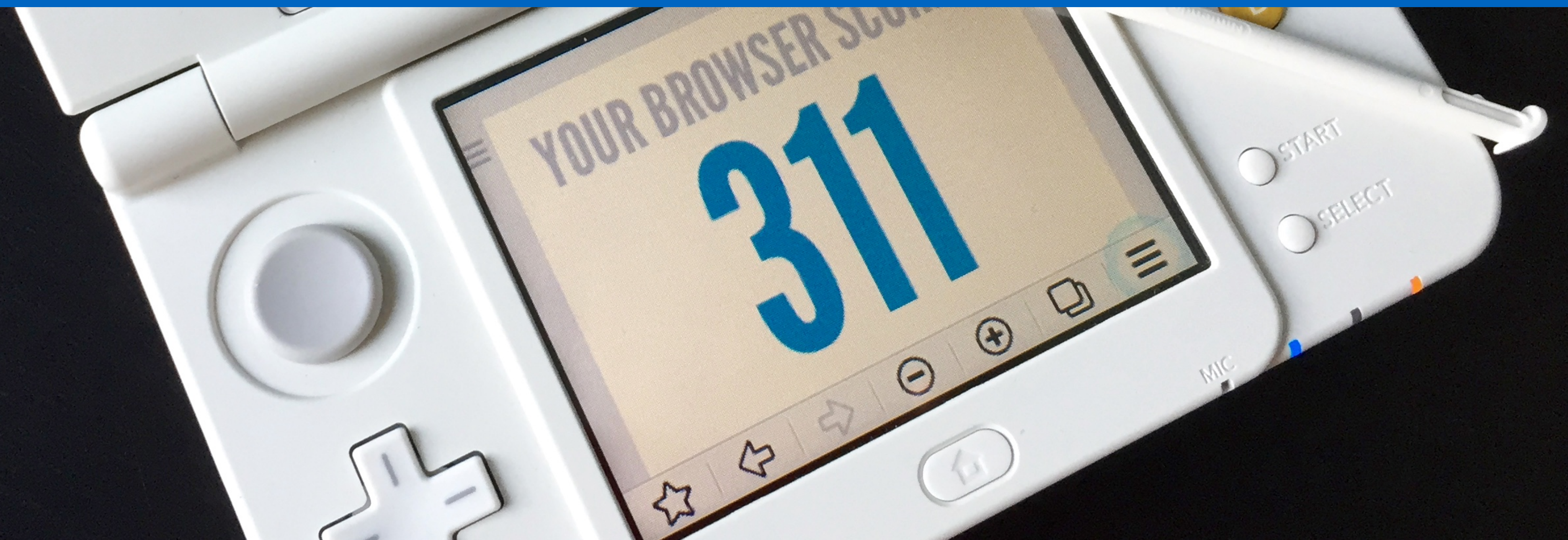


game consoles





portable game consoles







smart tvs





e-readers



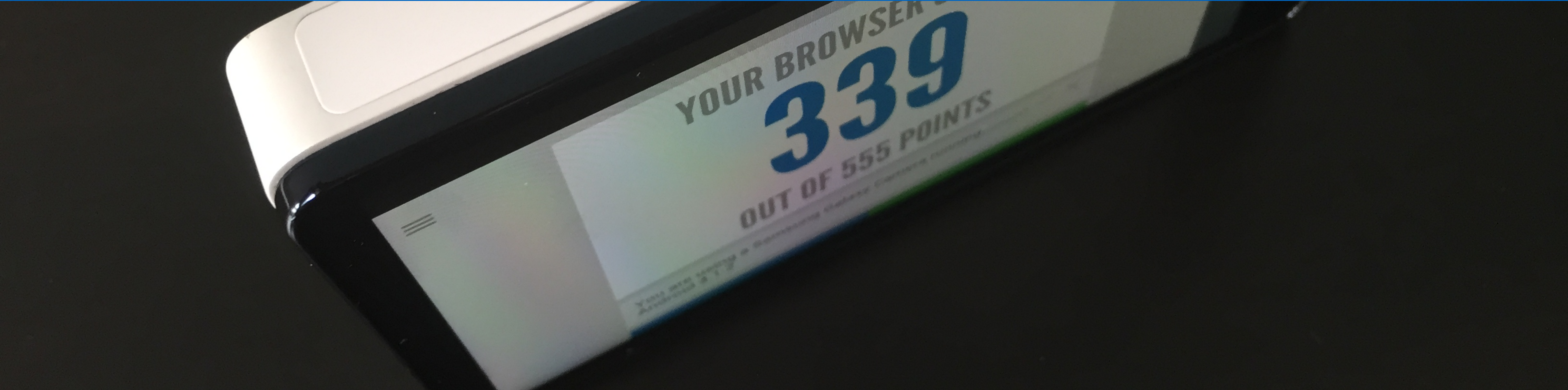


smartwatches





# photo cameras





YOUR BROWSER SCORES **216** OUT OF 555 POINTS

You are using a Tesla Model S

Correct? ✓ ✗

cars

### semantics

#### Parsing rules

5

<!DOCTYPE html> triggers standards mode	Yes	✓
HTML5 tokenizer	Yes	✓
HTML5 tree building	Yes	✓
<i>HTML5 defines rules for embedding SVG and MathML inside a regular HTML document. The following tests only check if the browser is following the HTML5 parsing rules for inline SVG and MathML, not if the browser can actually understand and render it.</i>		
Parsing inline SVG	Yes	✓

### multimedia

#### Video

0/35

video element	No	✗
Subtitles	No	✗
Audio track selection	No	✗
Video track selection	No	✗
Poster images	No	✗
Codec detection	No	✗
<b>Advanced</b>		
DRM support	No	✗



**what do they have in common?**









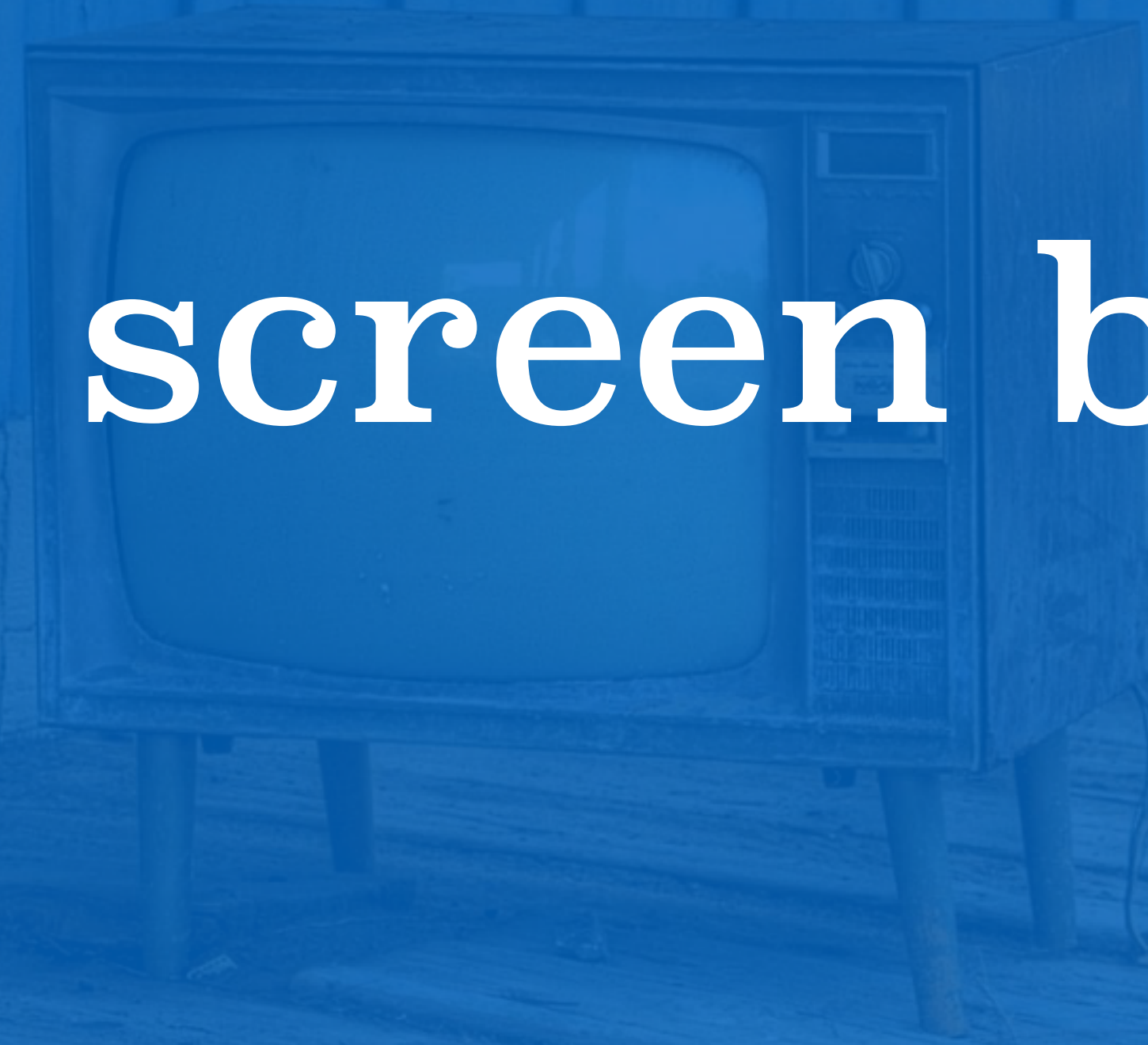


A vintage television set with a wooden cabinet and four legs sits on a wooden deck. In the background, there is a wooden door and a metal utility box. The entire image is covered with a semi-transparent blue overlay.

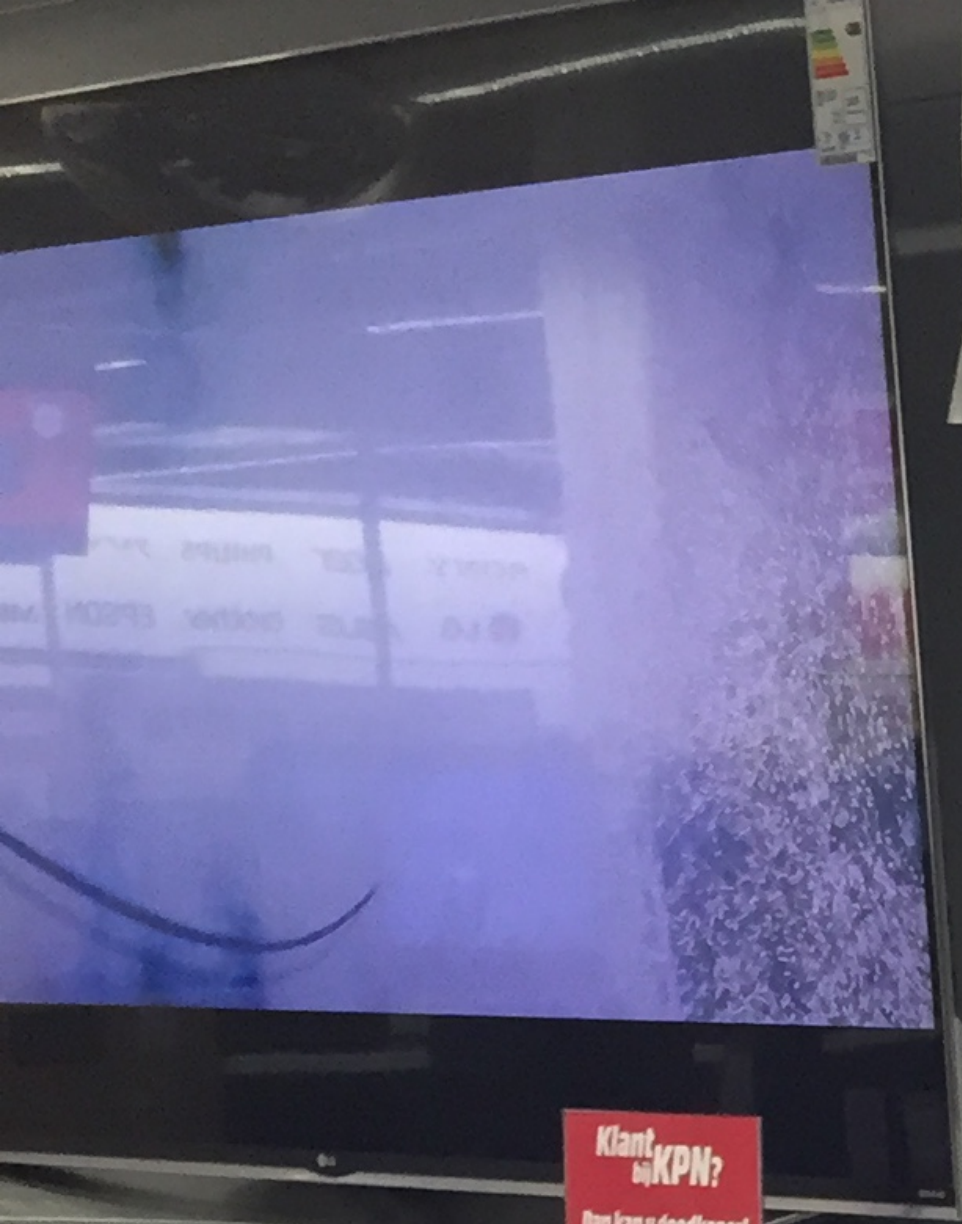
smart tvs, set-top boxes  
and consoles



“big screen browsers”



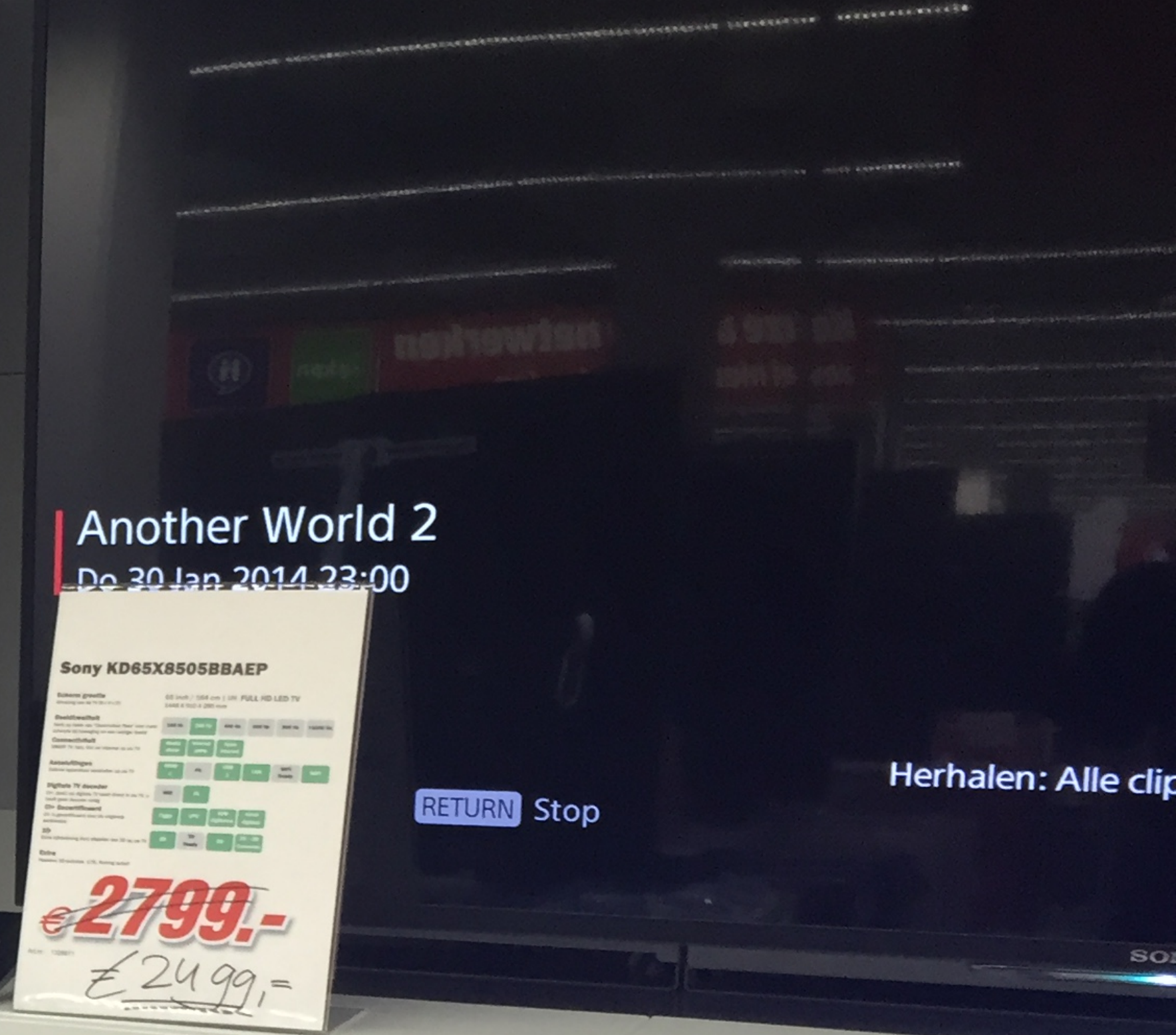




Klant  
KPN?  
Dan kan u goedkoop!



SONY  
KDL-8500WS  
85 inch LED TV  
3499.-  
MediaMarkt



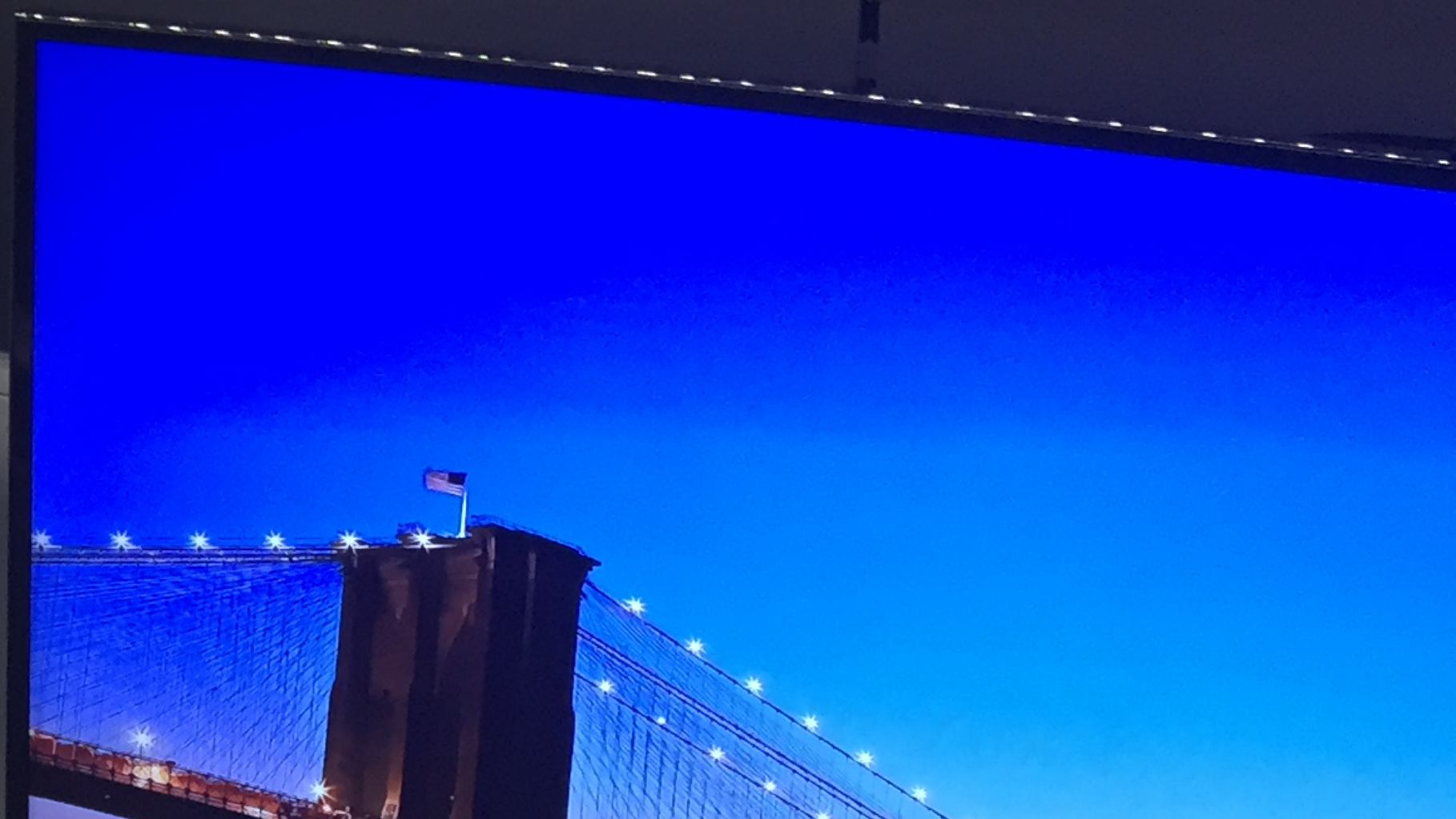
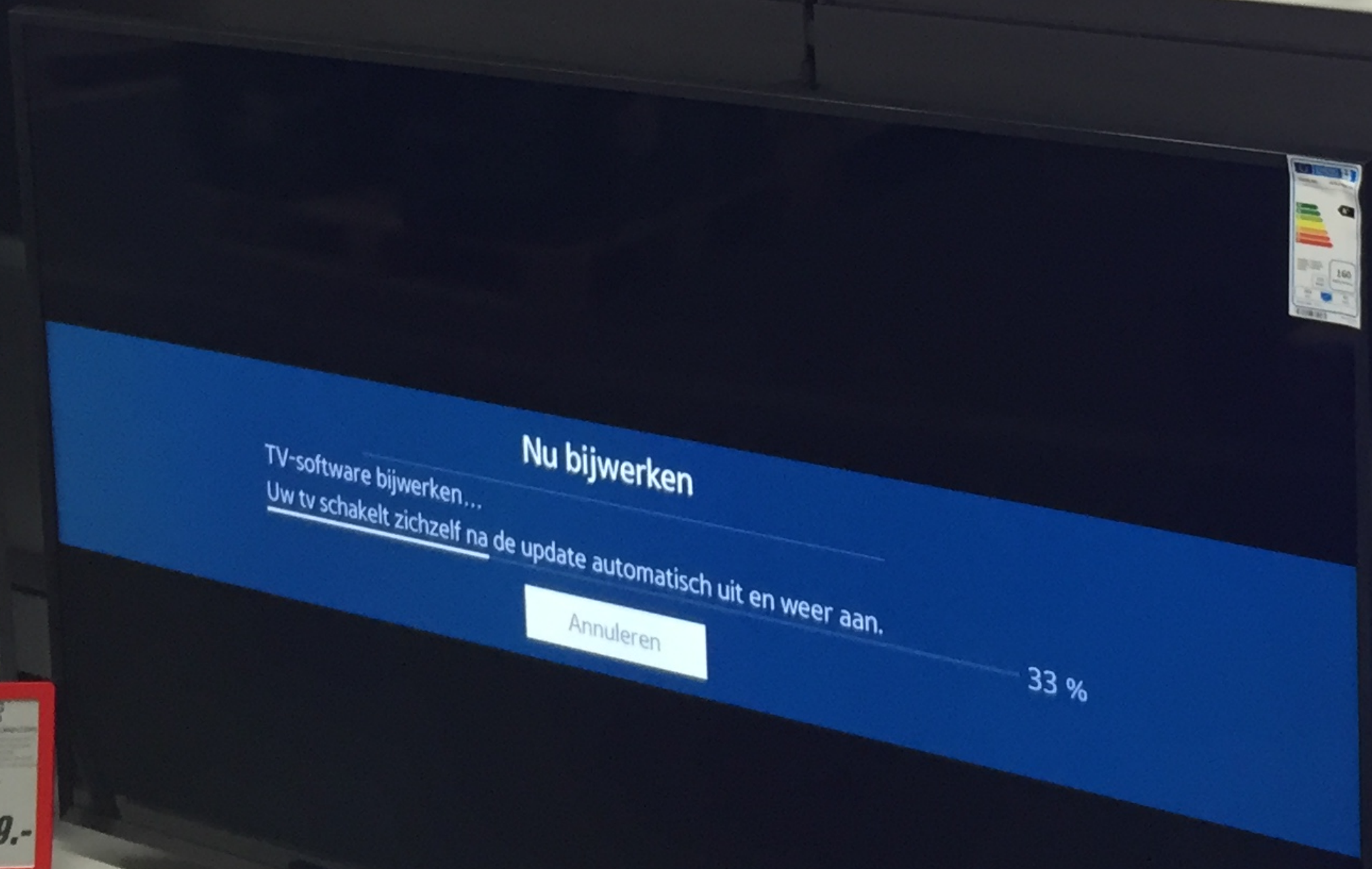
Sony KD65X8505BBAEP

Kenmerk	Kenmerk	Kenmerk	Kenmerk	Kenmerk
Model	65 inch	LED	Full HD	LED
Resolutie	1920 x 1080	LED	Full HD	LED
Kenmerk	LED	Full HD	LED	LED
Kenmerk	LED	Full HD	LED	LED
Kenmerk	LED	Full HD	LED	LED
Kenmerk	LED	Full HD	LED	LED
Kenmerk	LED	Full HD	LED	LED
Kenmerk	LED	Full HD	LED	LED
Kenmerk	LED	Full HD	LED	LED
Kenmerk	LED	Full HD	LED	LED

**€2799.-**  
€2499,-



SAMSUNG  
UE85JU7000  
85 inch LED TV  
3699.-  
MediaMarkt

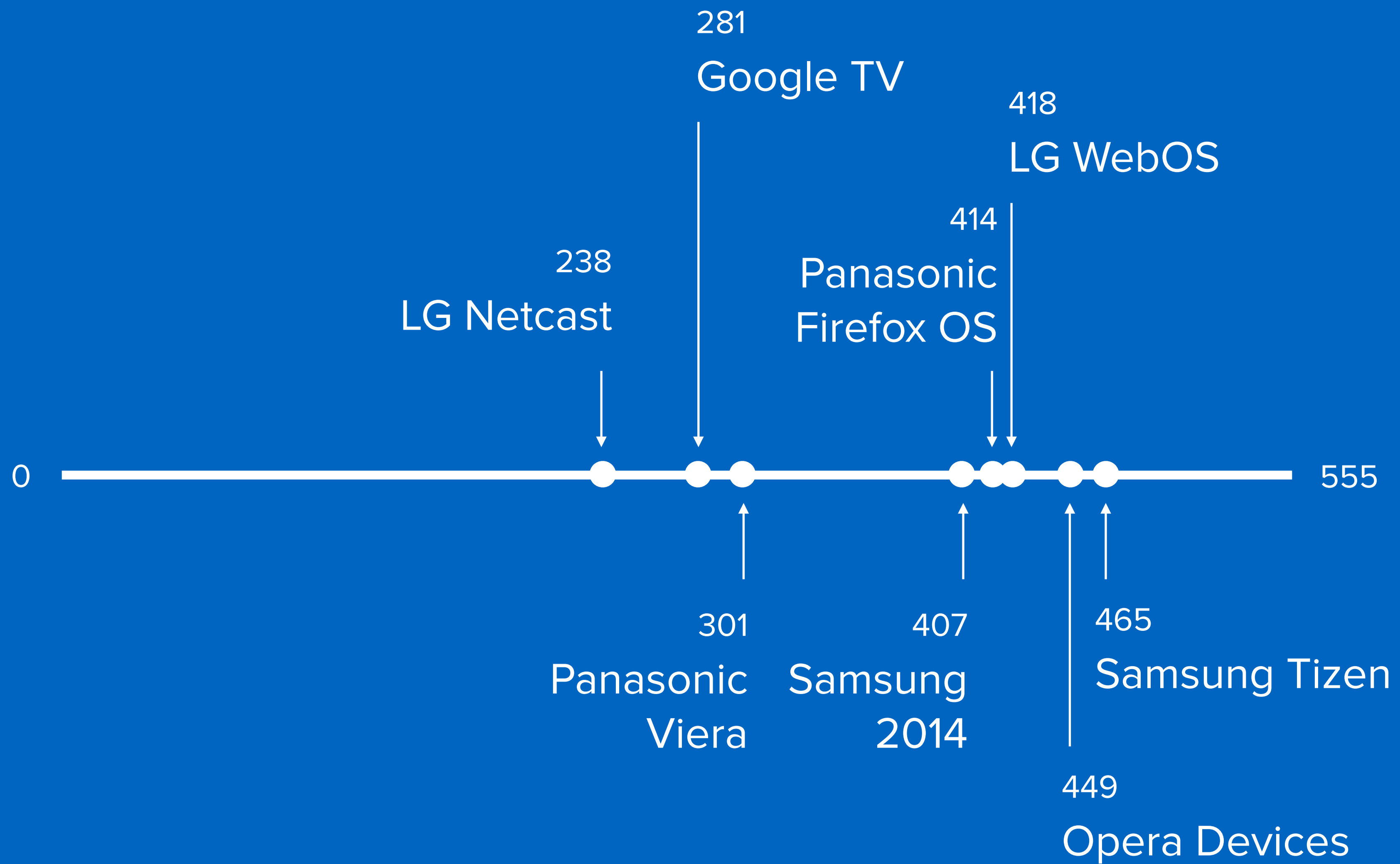




**television browsers are pretty good**

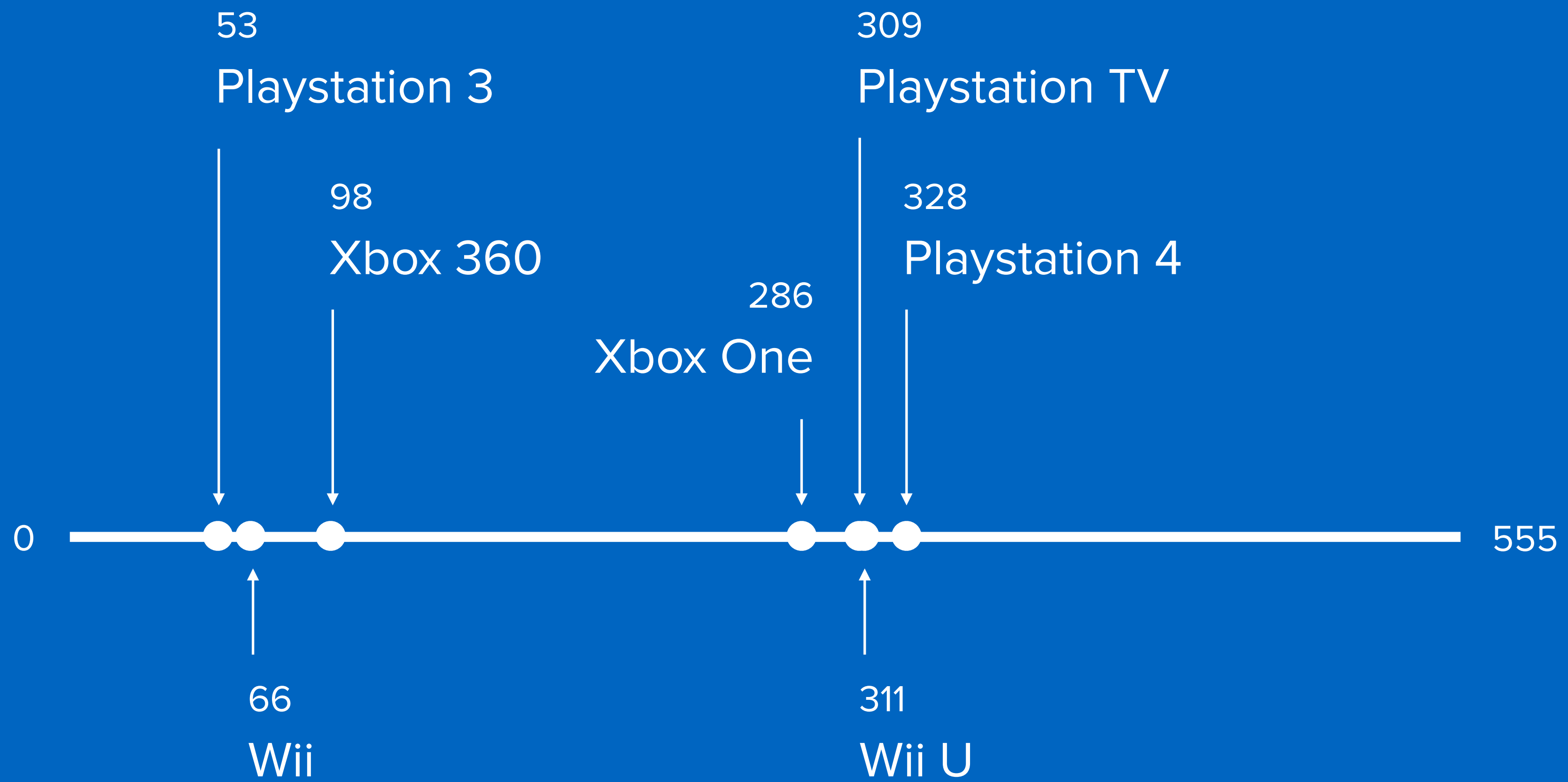
**the last generation of television sets use  
operating systems that originate from mobile**





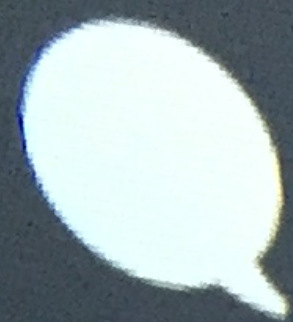
smart tv results on [html5test.com](http://html5test.com)





console results on [html5test.com](http://html5test.com)





Out of Memory. App is closed.



1

control



**the biggest challenge of  
of television browsers**



navigation  
(without mouse or touchscreen)

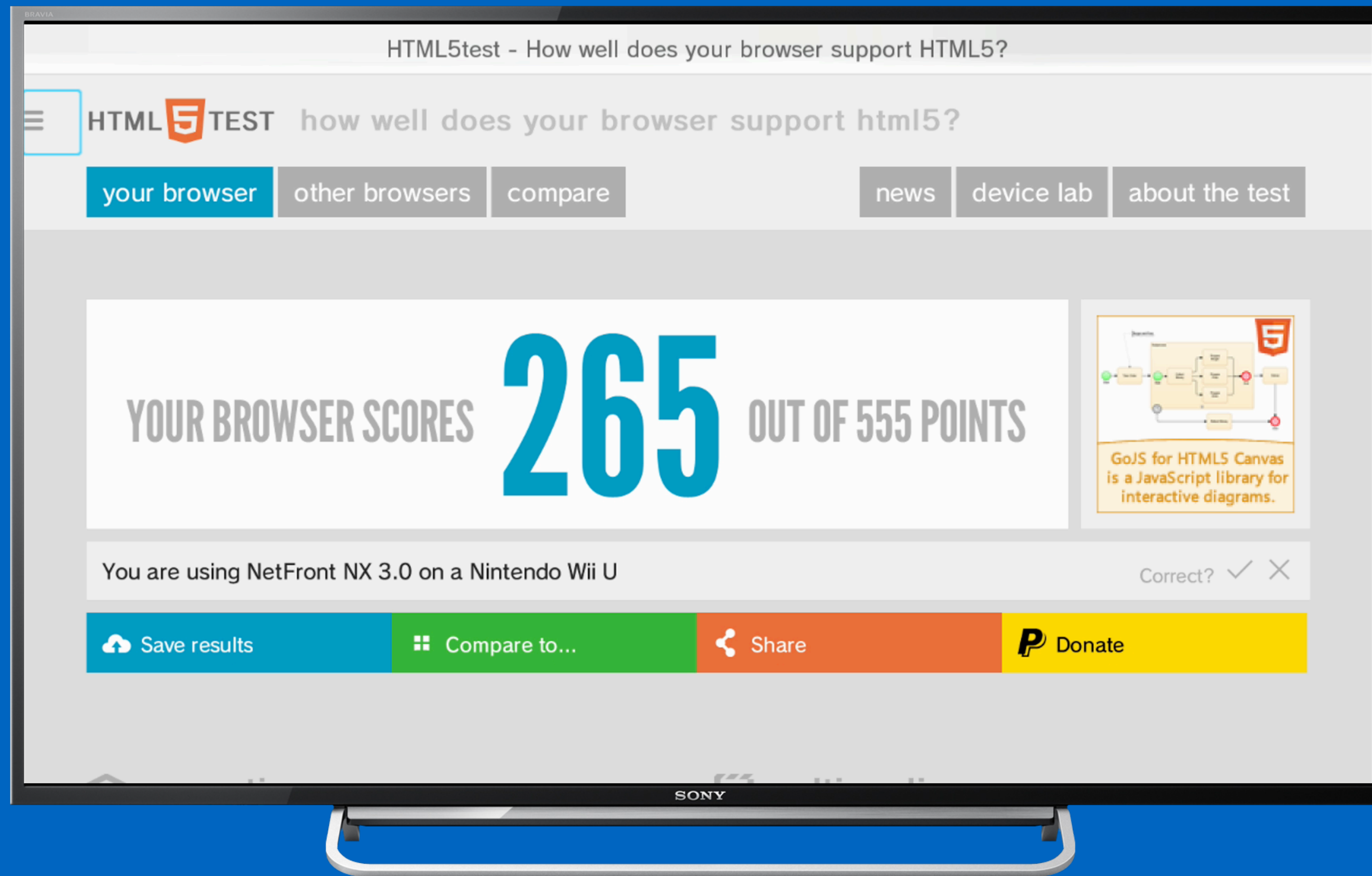




d-pad







navigation with the d-pad



**but it can be worse:  
moving the cursor with the arrow keys**



**alternatives**





analog controllers





remotes with trackpad







remotes with airmouse





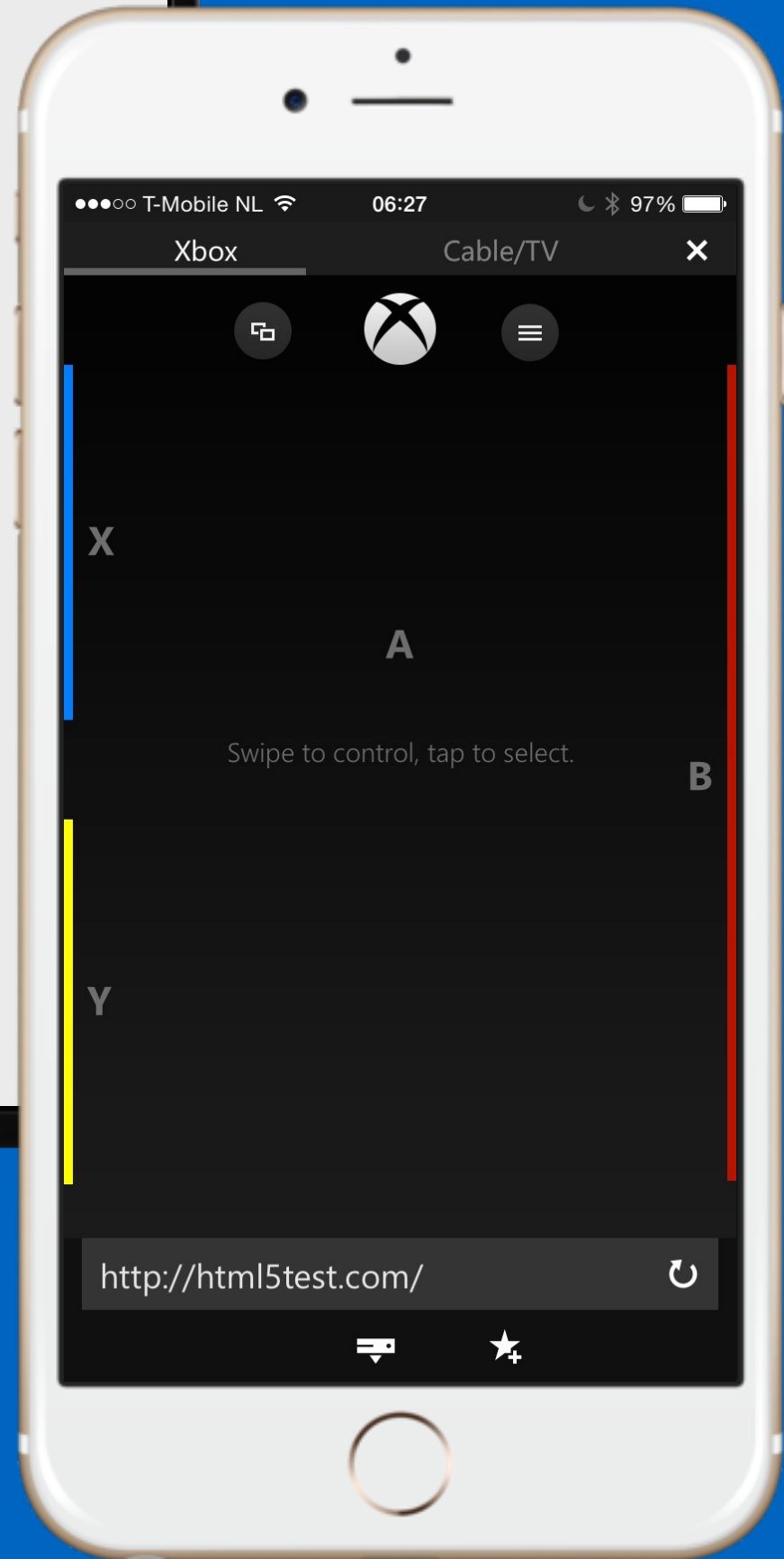
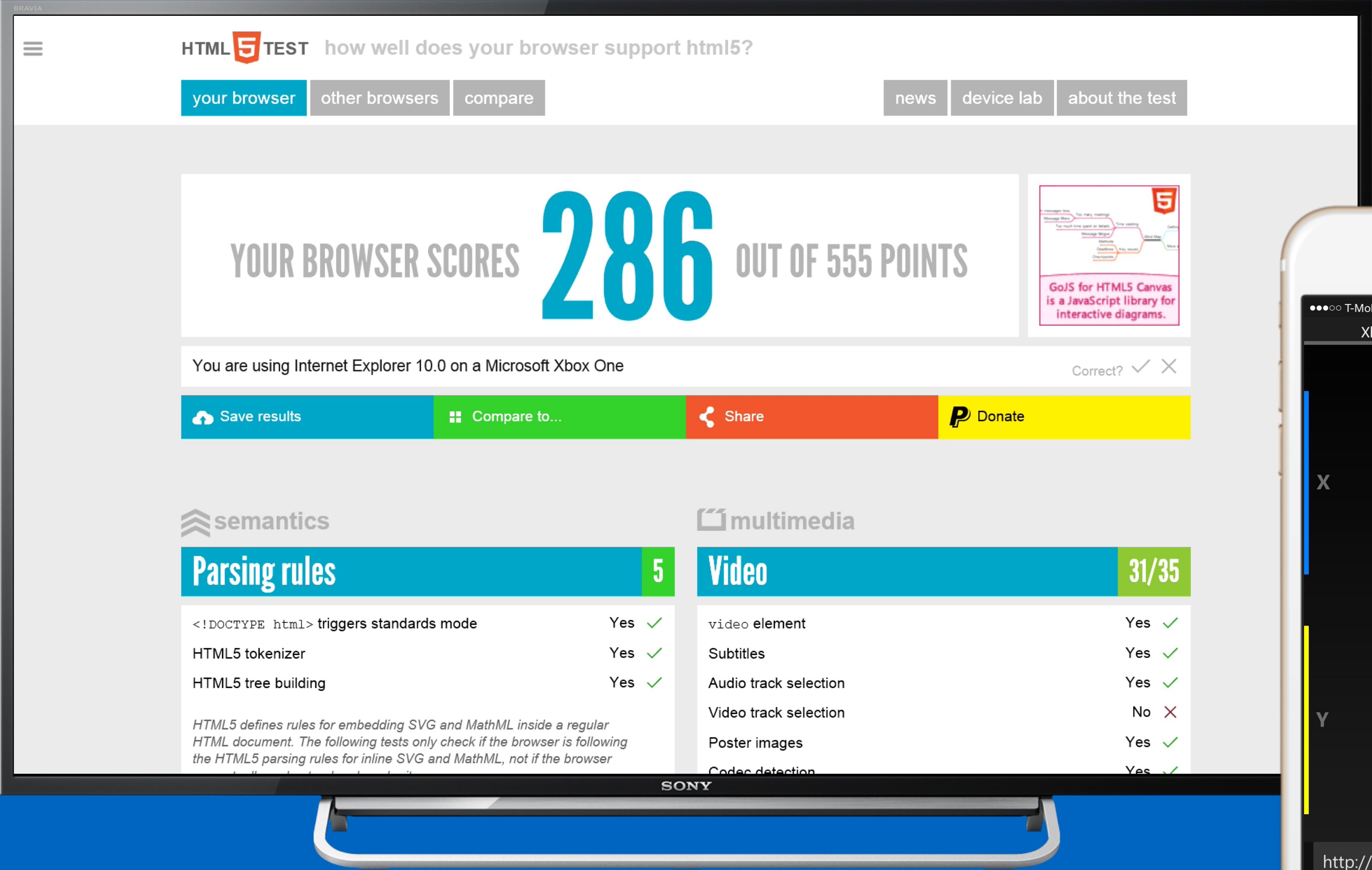


second screen

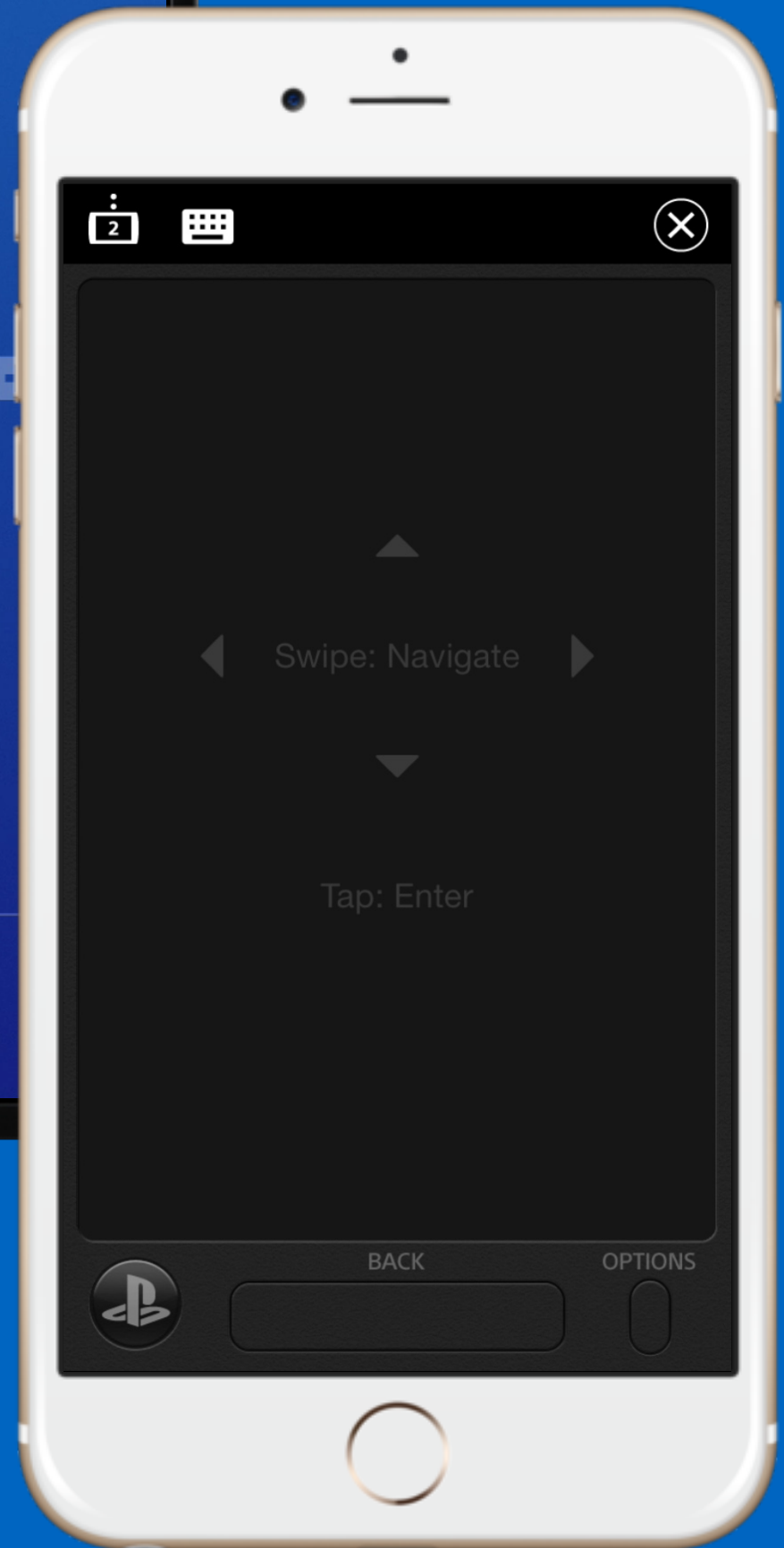
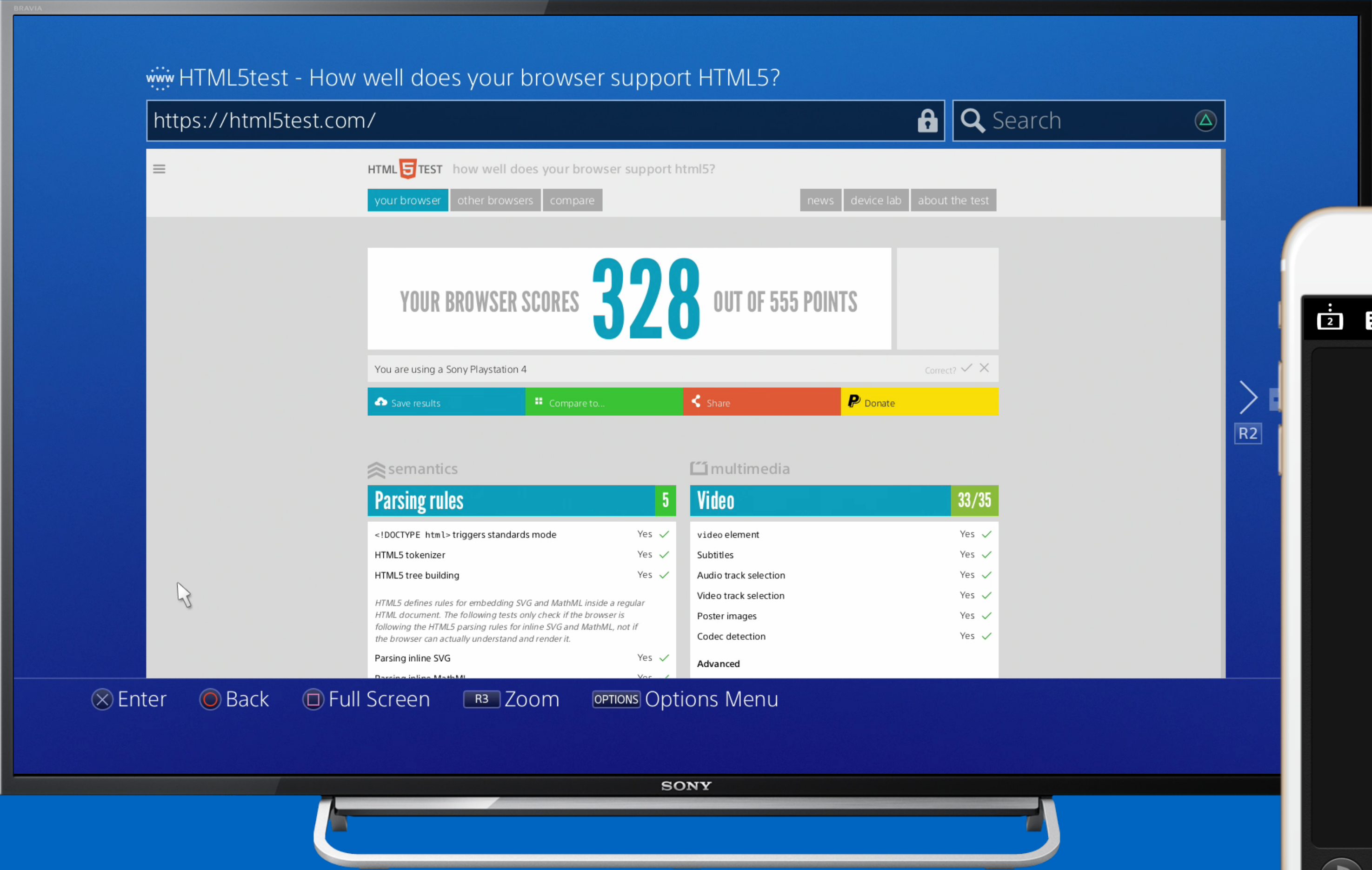


**many manufacturers also create apps for  
controlling the smart tv, console or set-top box**

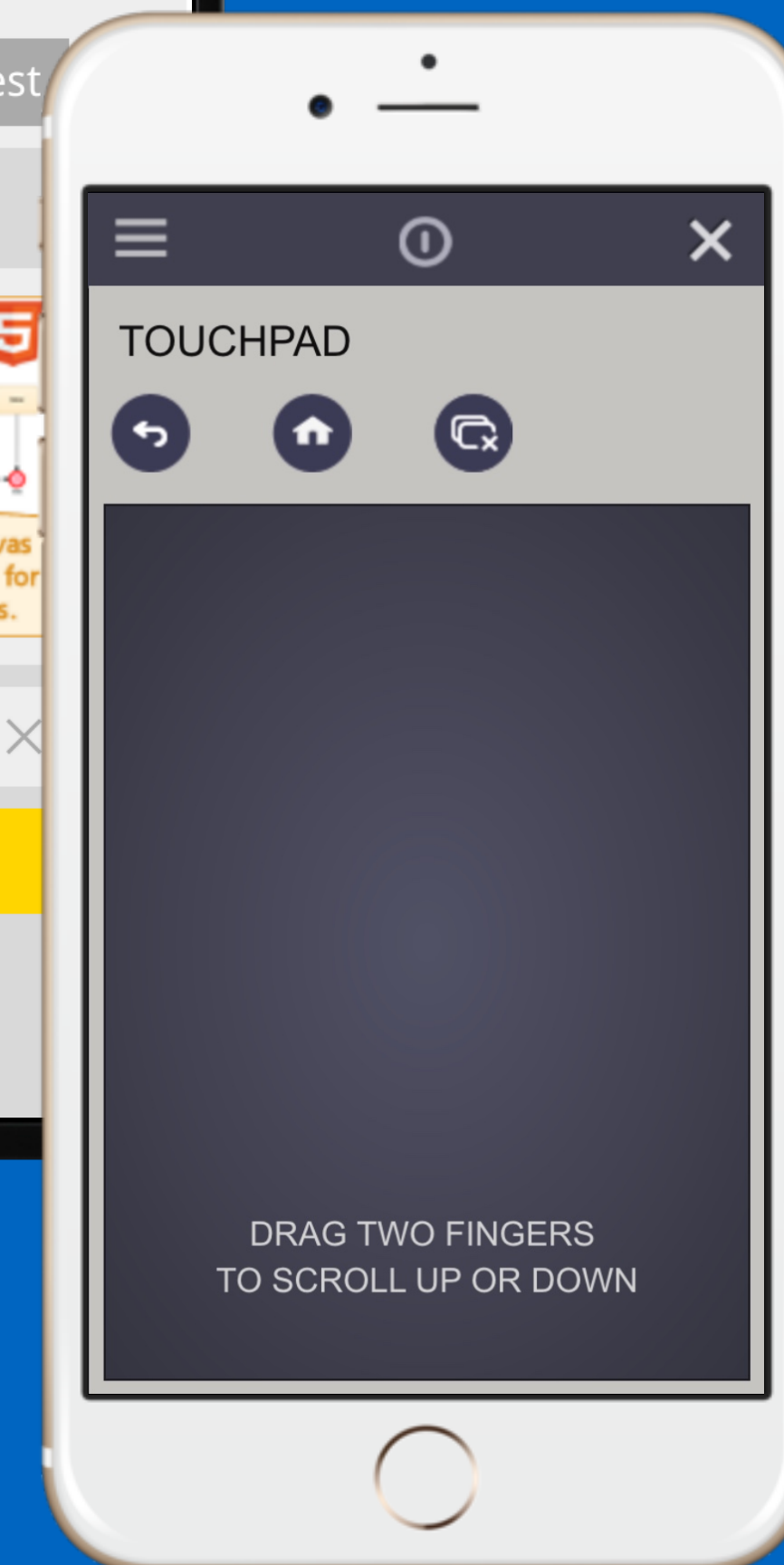
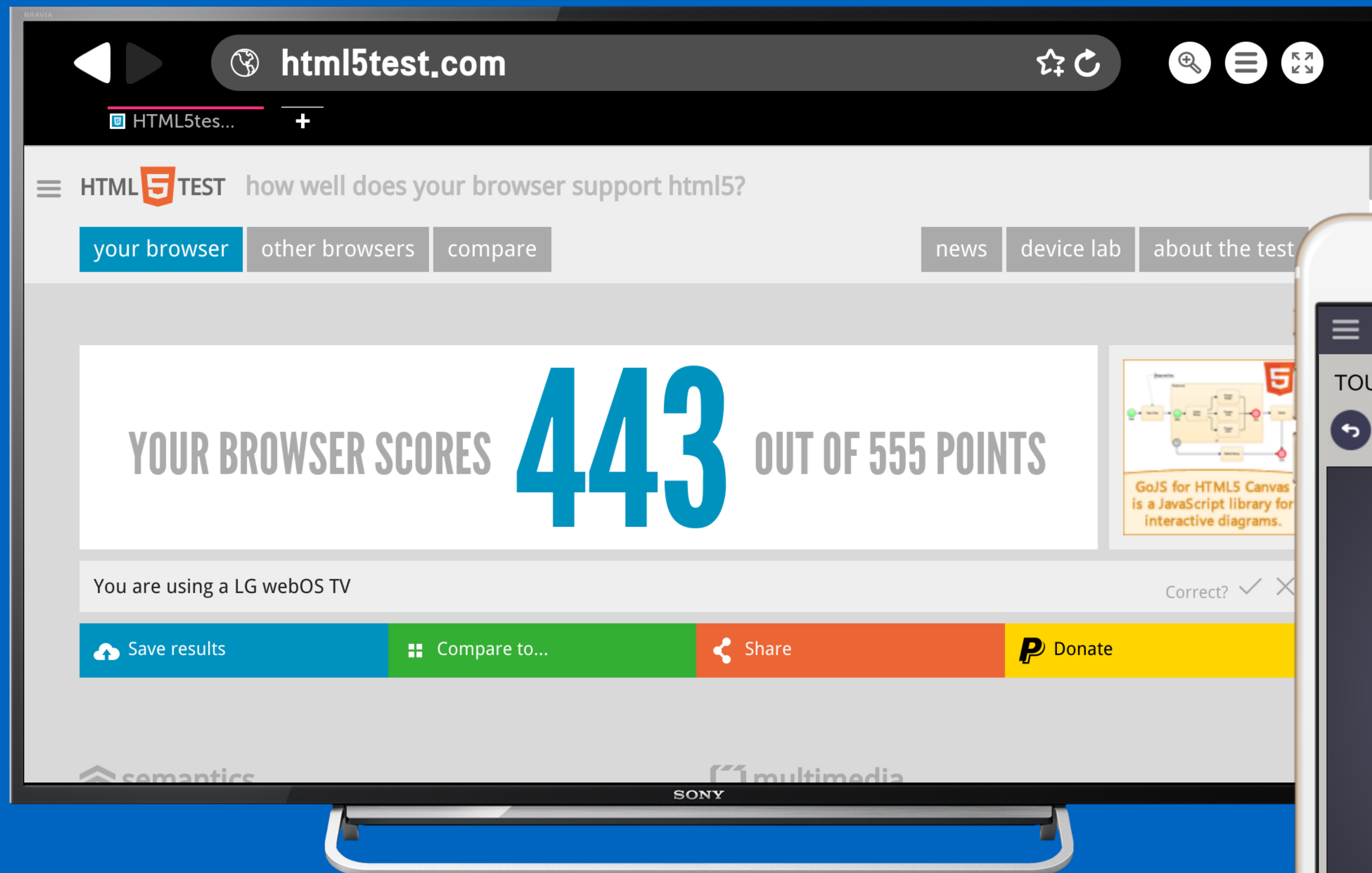














text input  
(without keyboard)

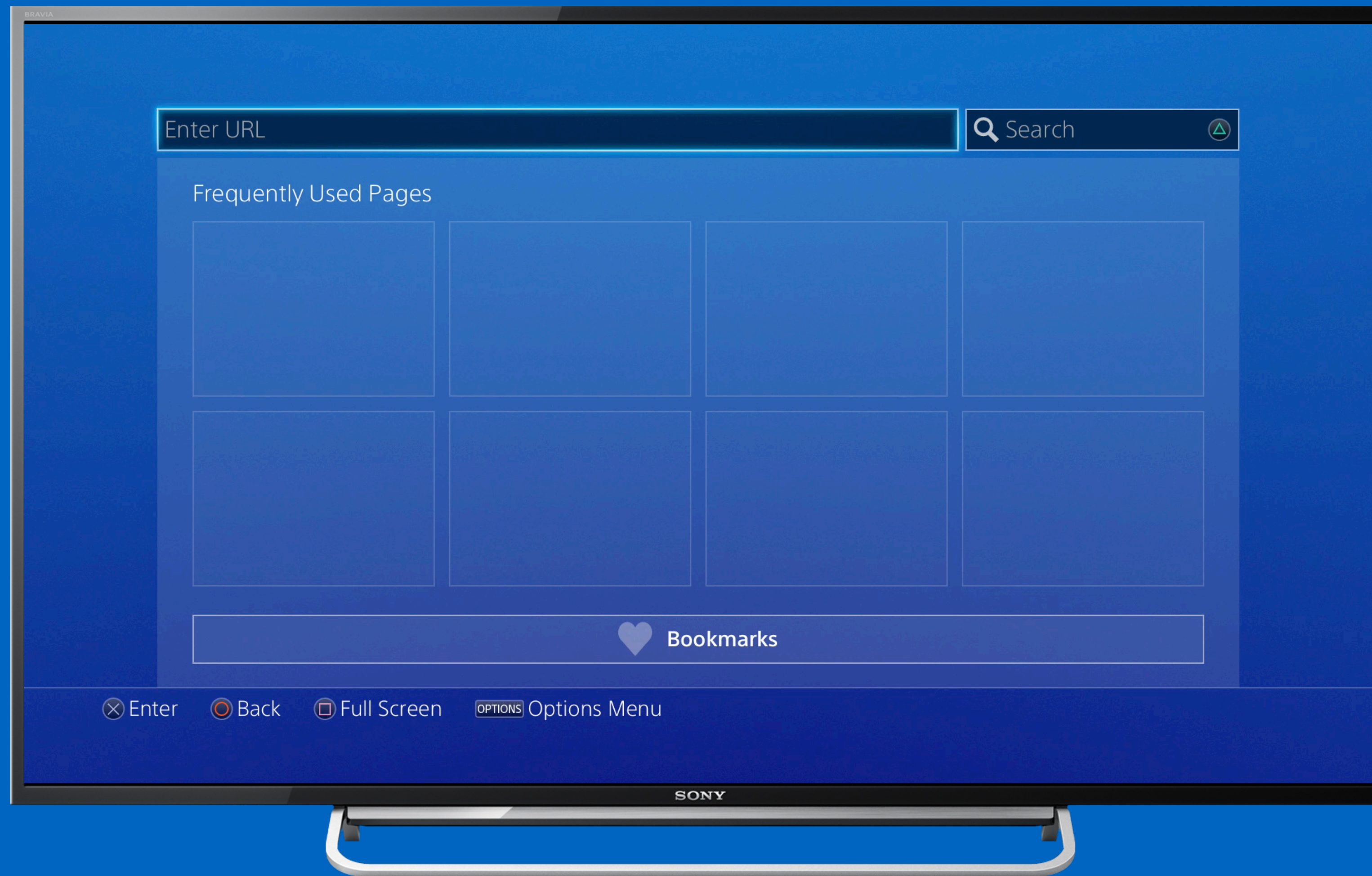




d-pads







text input with the d-pad



**alternatives**



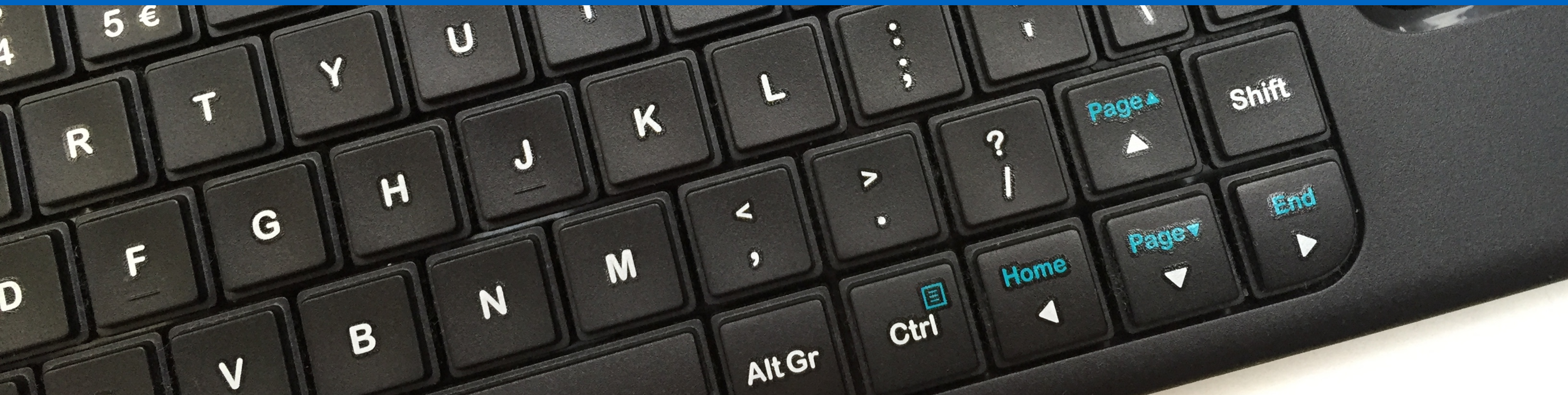


remotes with keyboards





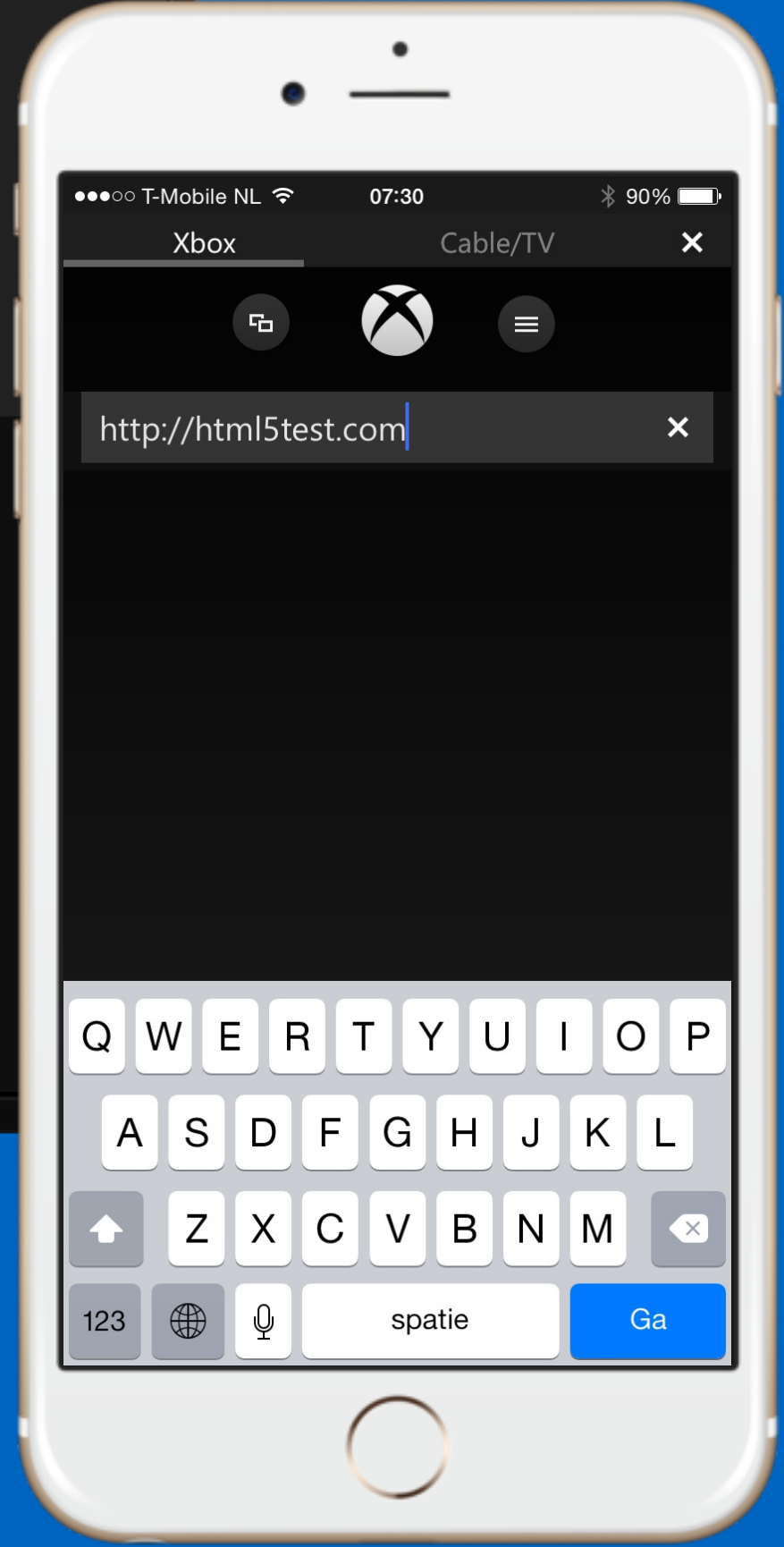
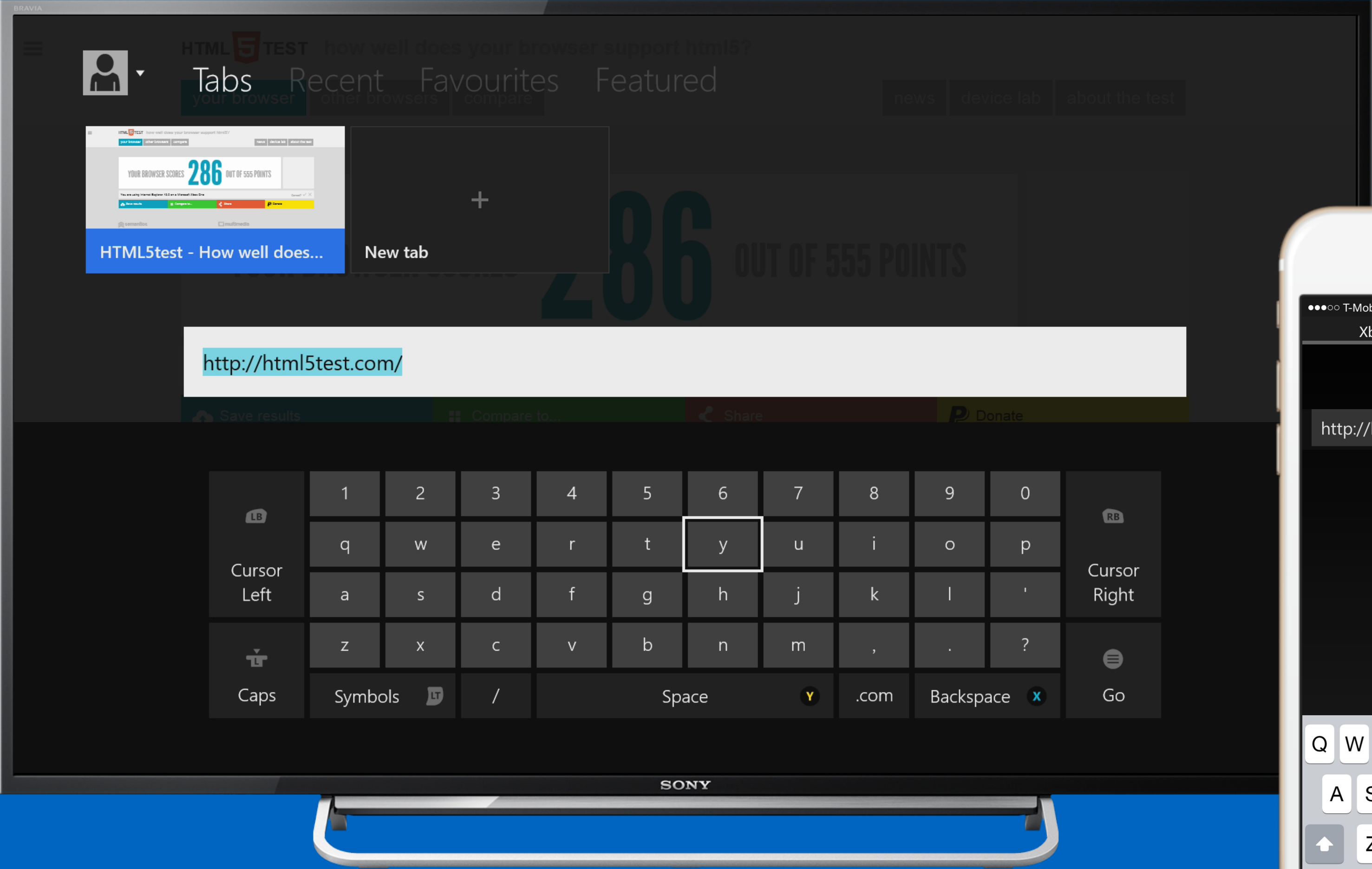
# wireless keyboards



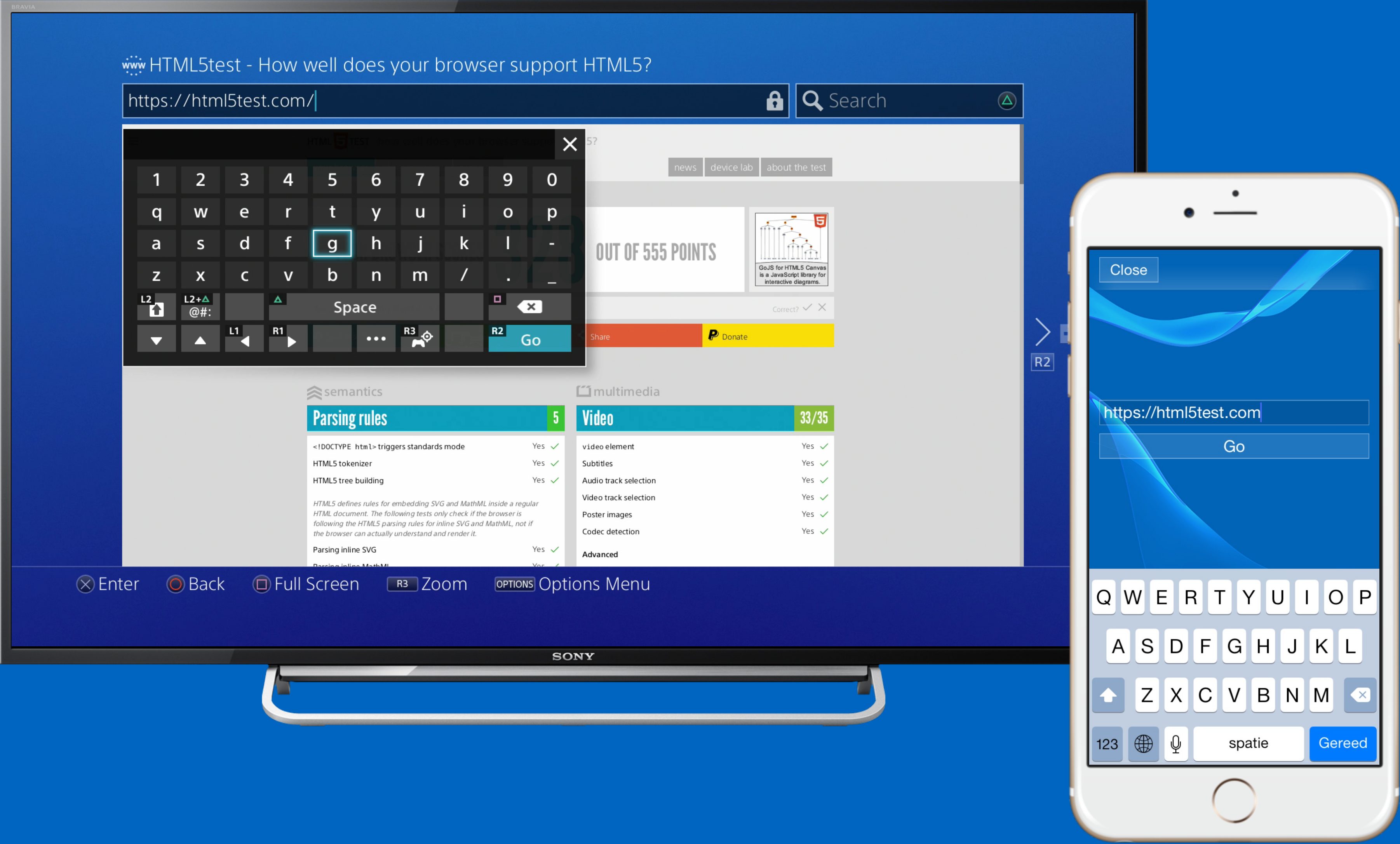


**and apps**







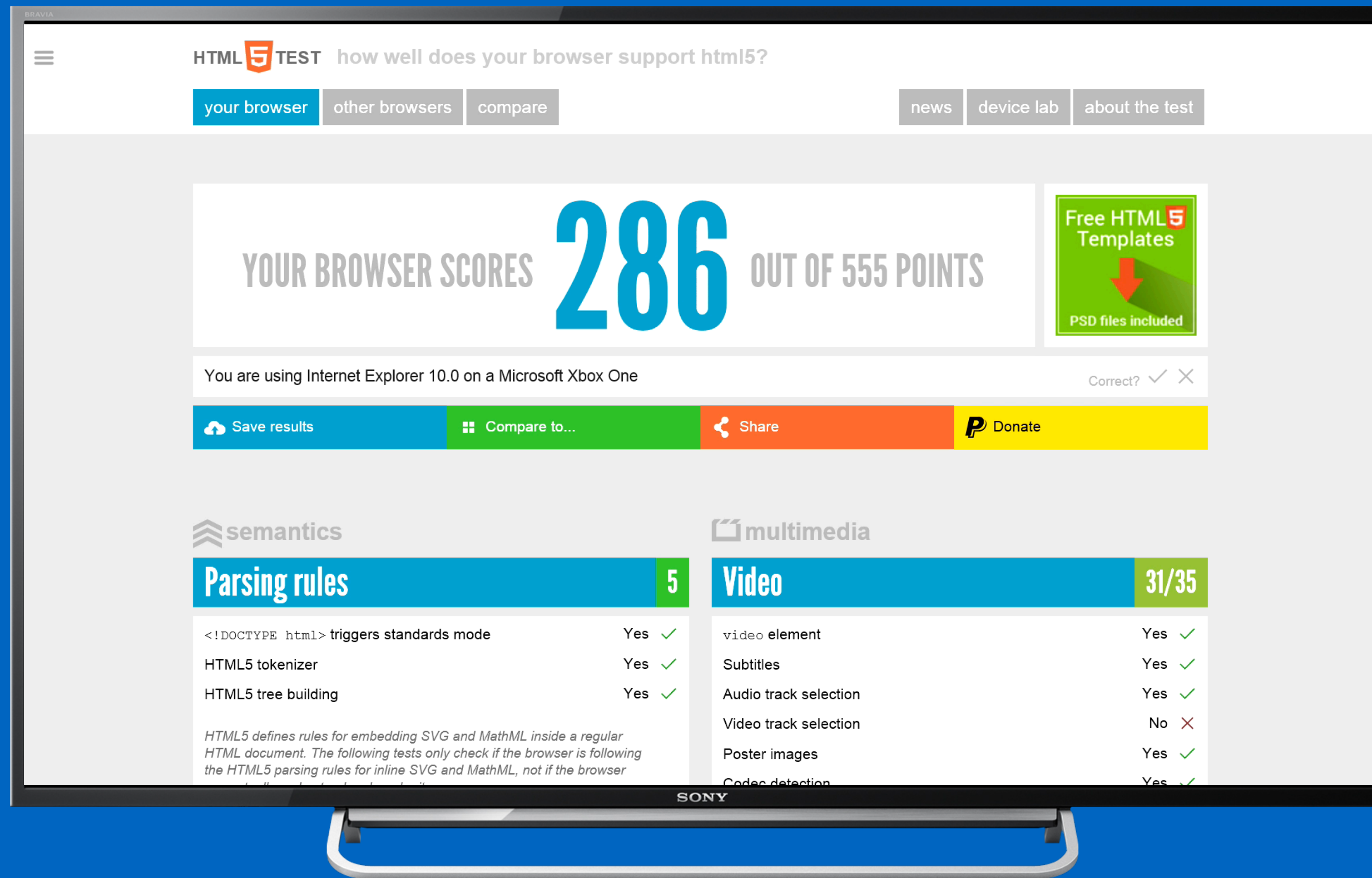




# gesture control

(throw your hands up in the air,  
and wave 'em like you just don't care)





navigation with gesture control



can we control these input methods  
directly from javascript?



the d-pad  
maybe



# 1 keyboard events

```
window.addEventListener("keypress", function(e) {  
    e.preventDefault(); // no navigation  
    ...  
});
```



the gamepad  
maybe



# 1 the gamepad api

```
var gamepads = navigator.getGamepads();  
for (var i = 0; i < gamepads.length; i++) {  
    ...  
}
```



2

## wii u api

```
window.setInterval(function() {  
    var state = window.wiiu.gamepad.update();  
    ...  
}, 100);
```



the webcam

no\*



gestures

no\*



2

the difference between  
a television and a monitor



**overscan**

(let's make it a bit more complicated)



**due to historical reasons televisions will  
not show the borders of the image**



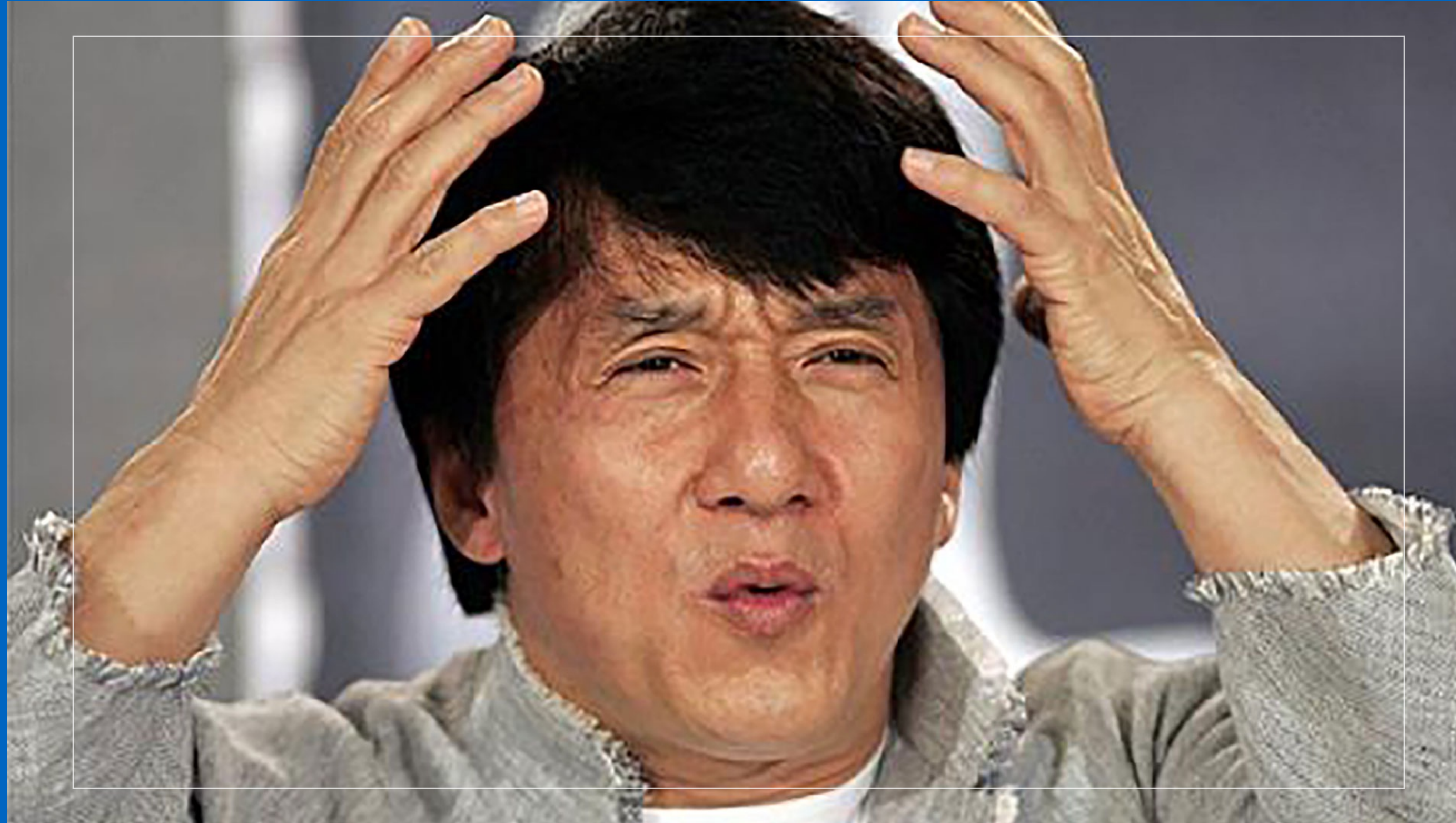
1920 pixels



**the television enlarges all images  
from the hdmi input by 5%**

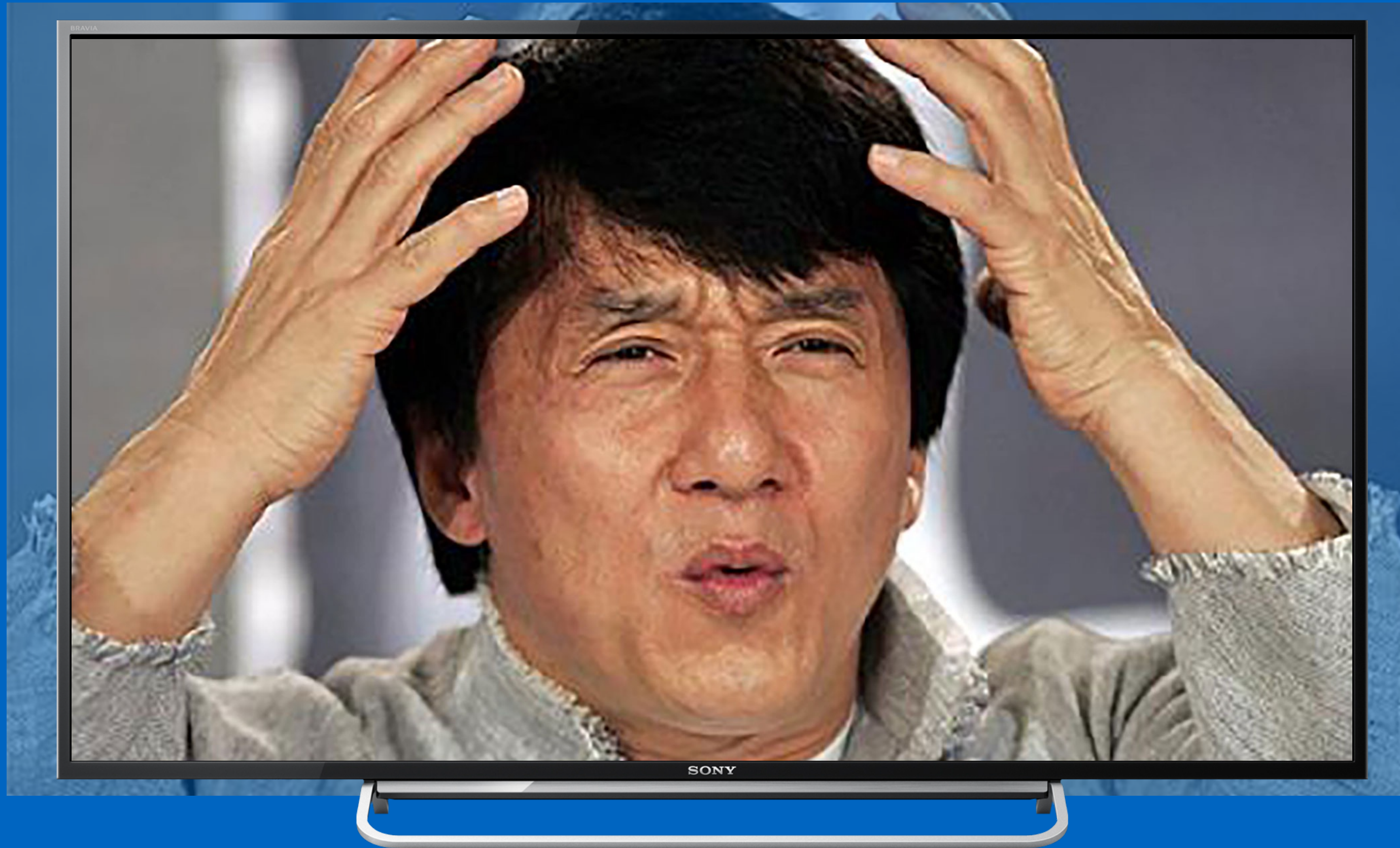


1920 pixels



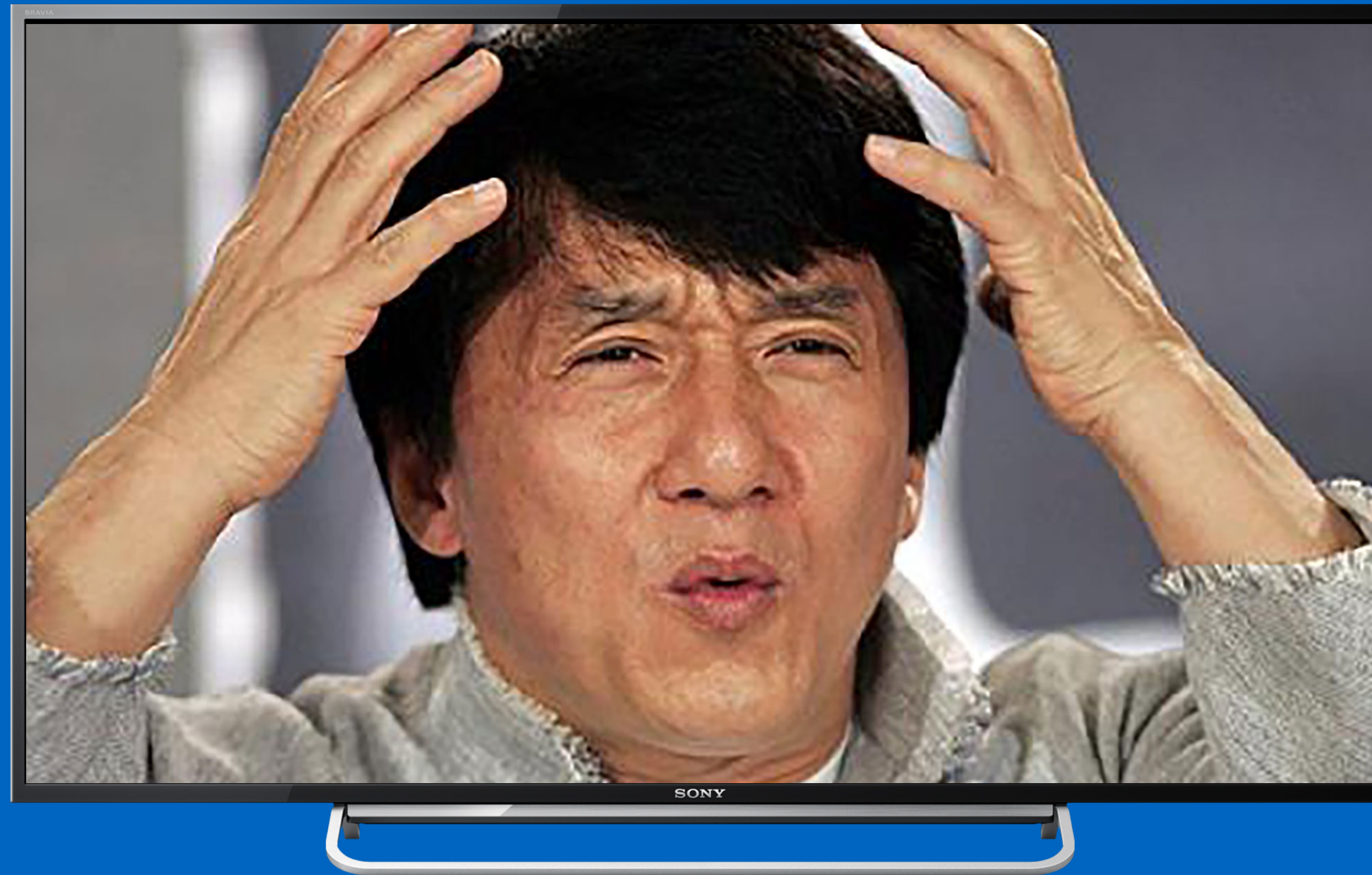
**the television enlarges all images  
from the hdmi input by 5%**





**the image is then cropped to  
1920 by 1080 pixels**

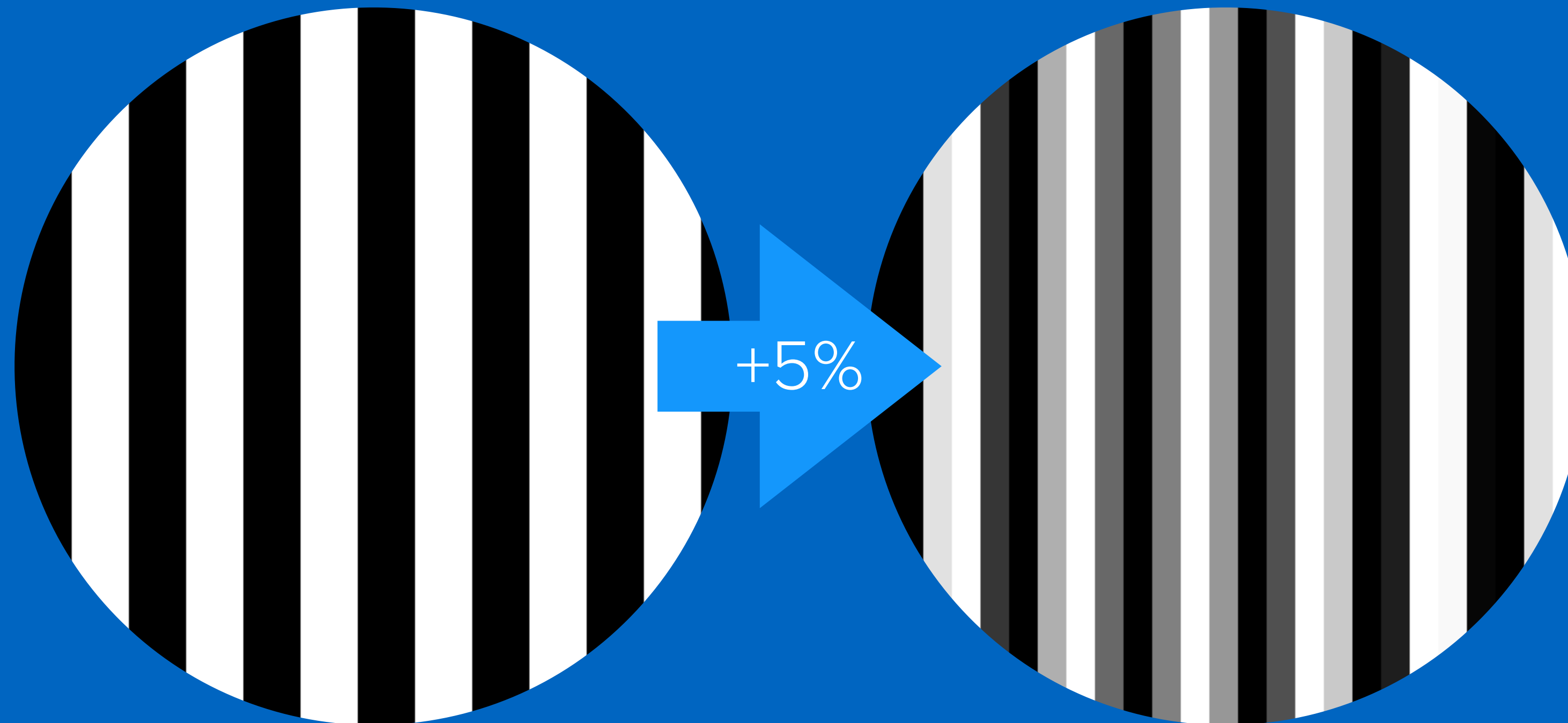




**the image is then cropped to  
1920 by 1080 pixels**



**overscan causes blurry output**



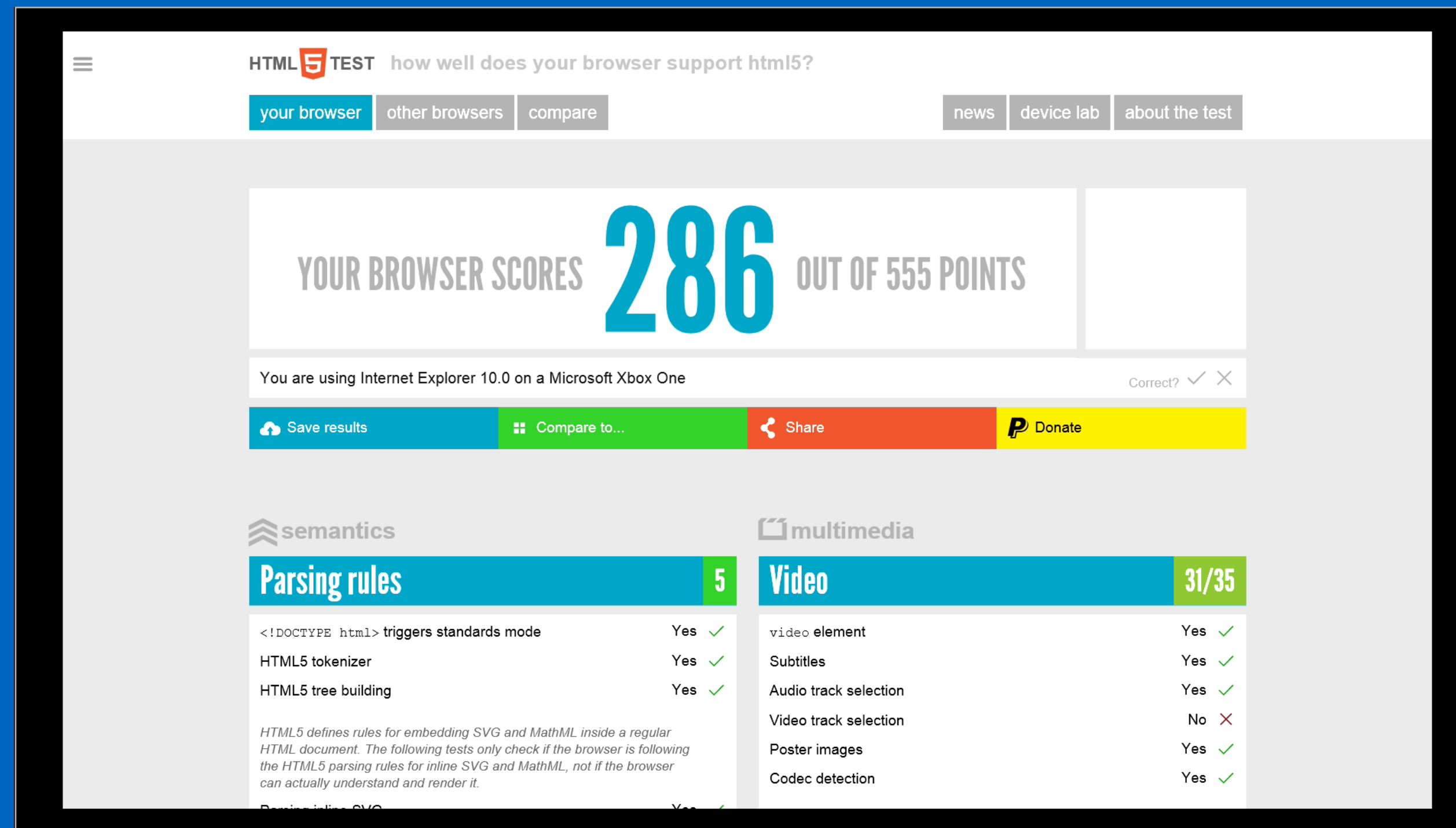


**solution 1**

**overscan correction**

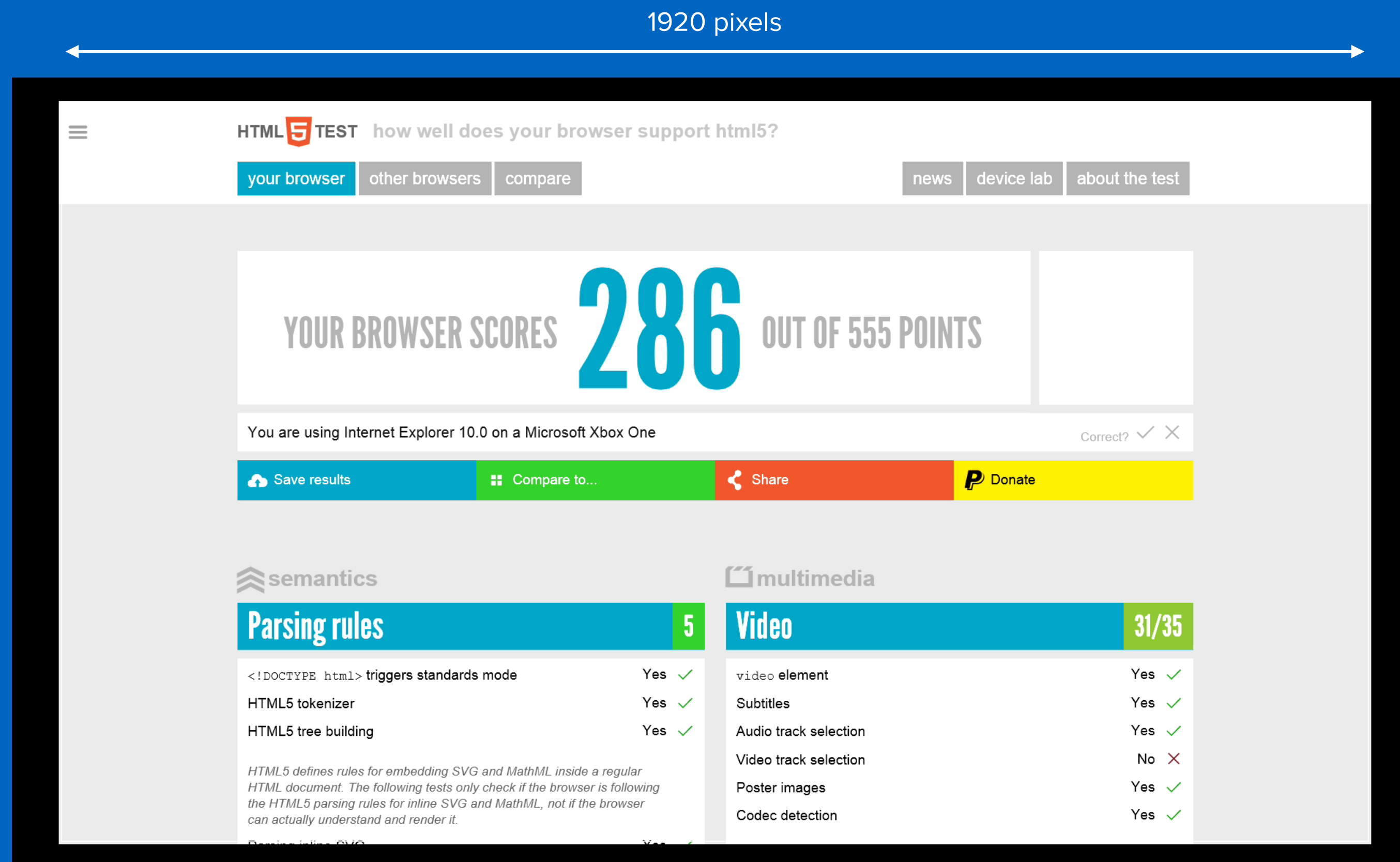


1920 pixels



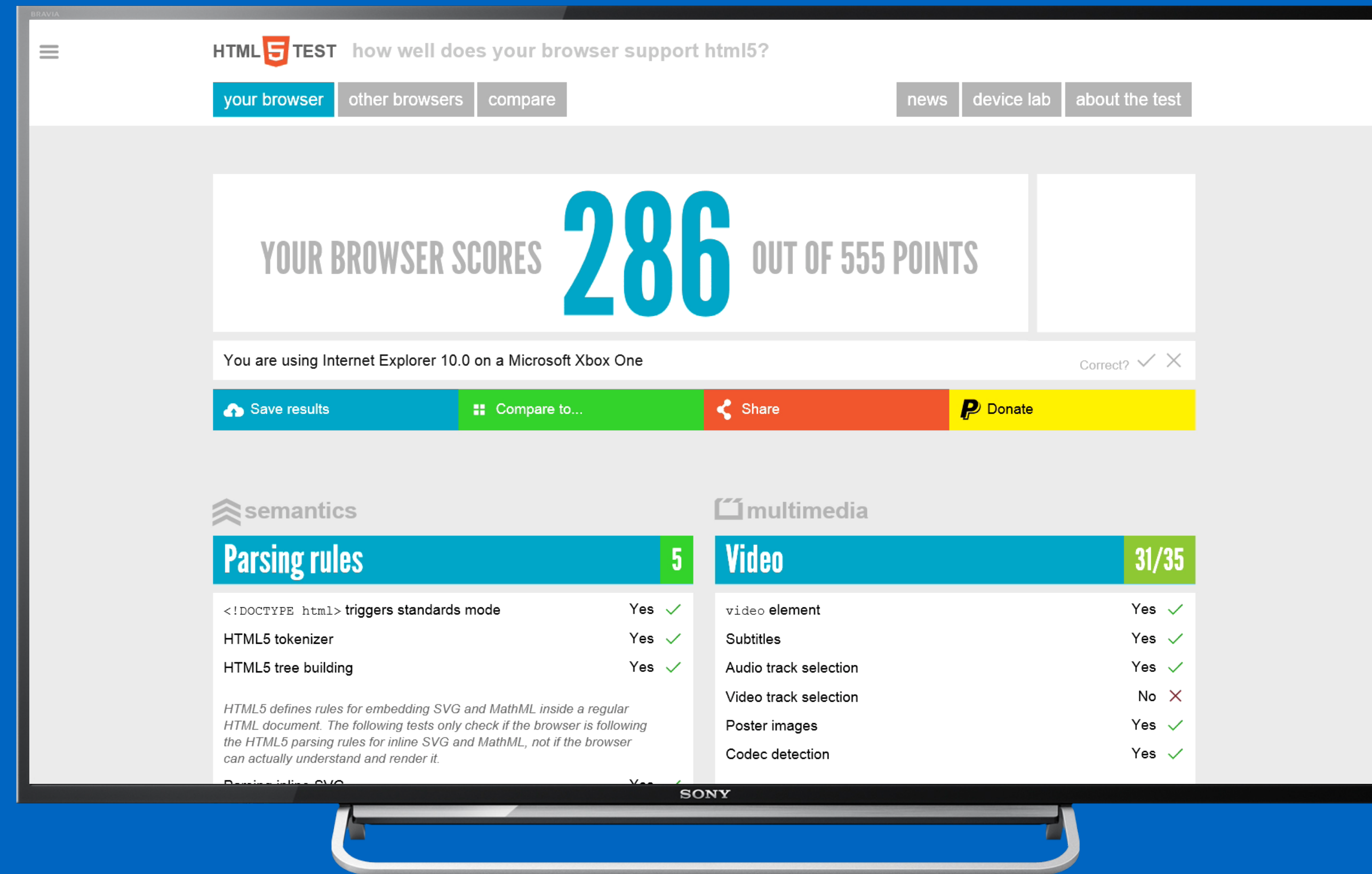
the browser does not use  
the edges of the image





the television will enlarge  
the image by 5%





and the content is now fully visible, the unused border is cropped out of the final image



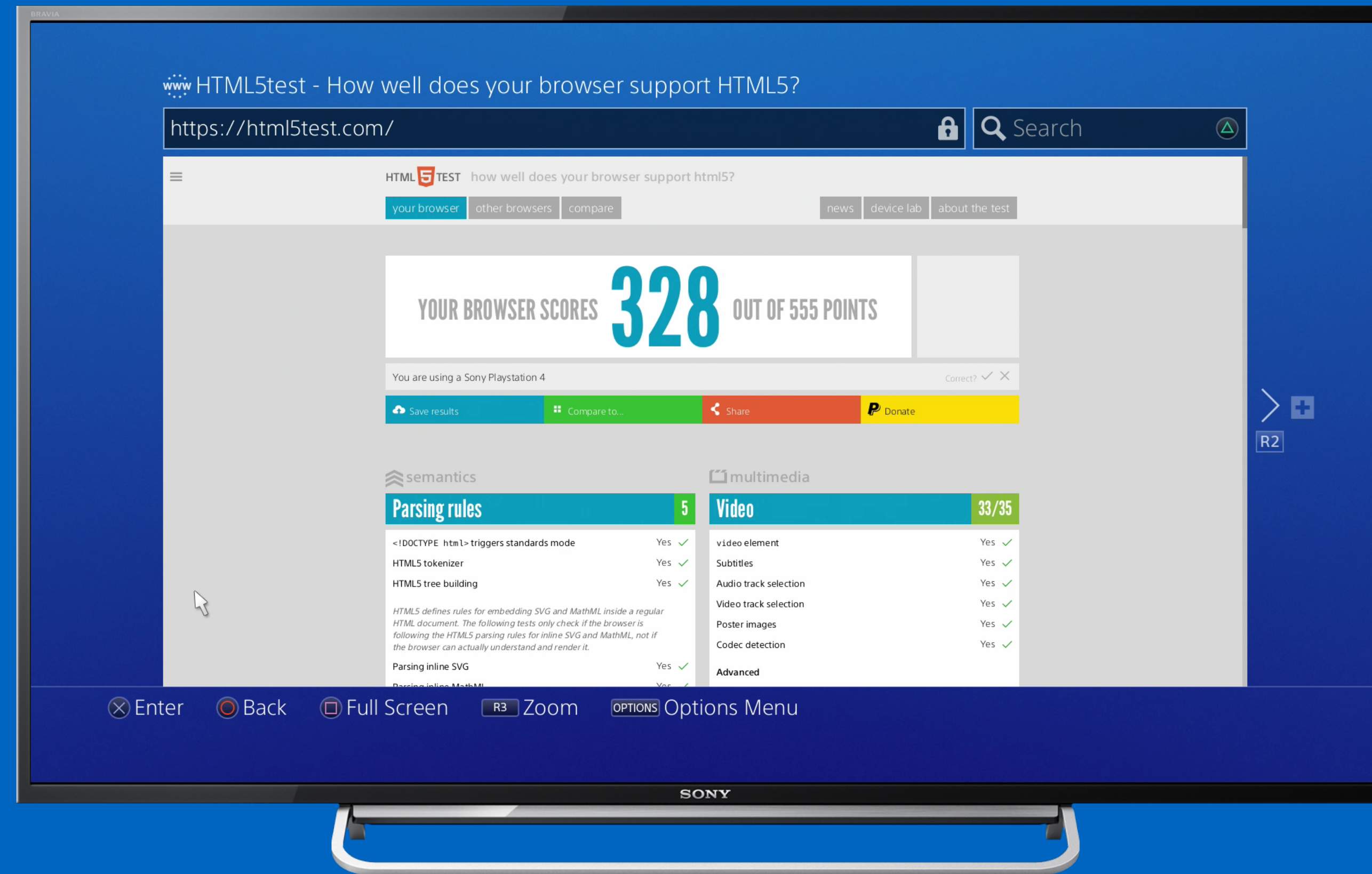
**but not every television set enlarges the  
image by exactly 5%, this can vary between  
manufacturers and models**





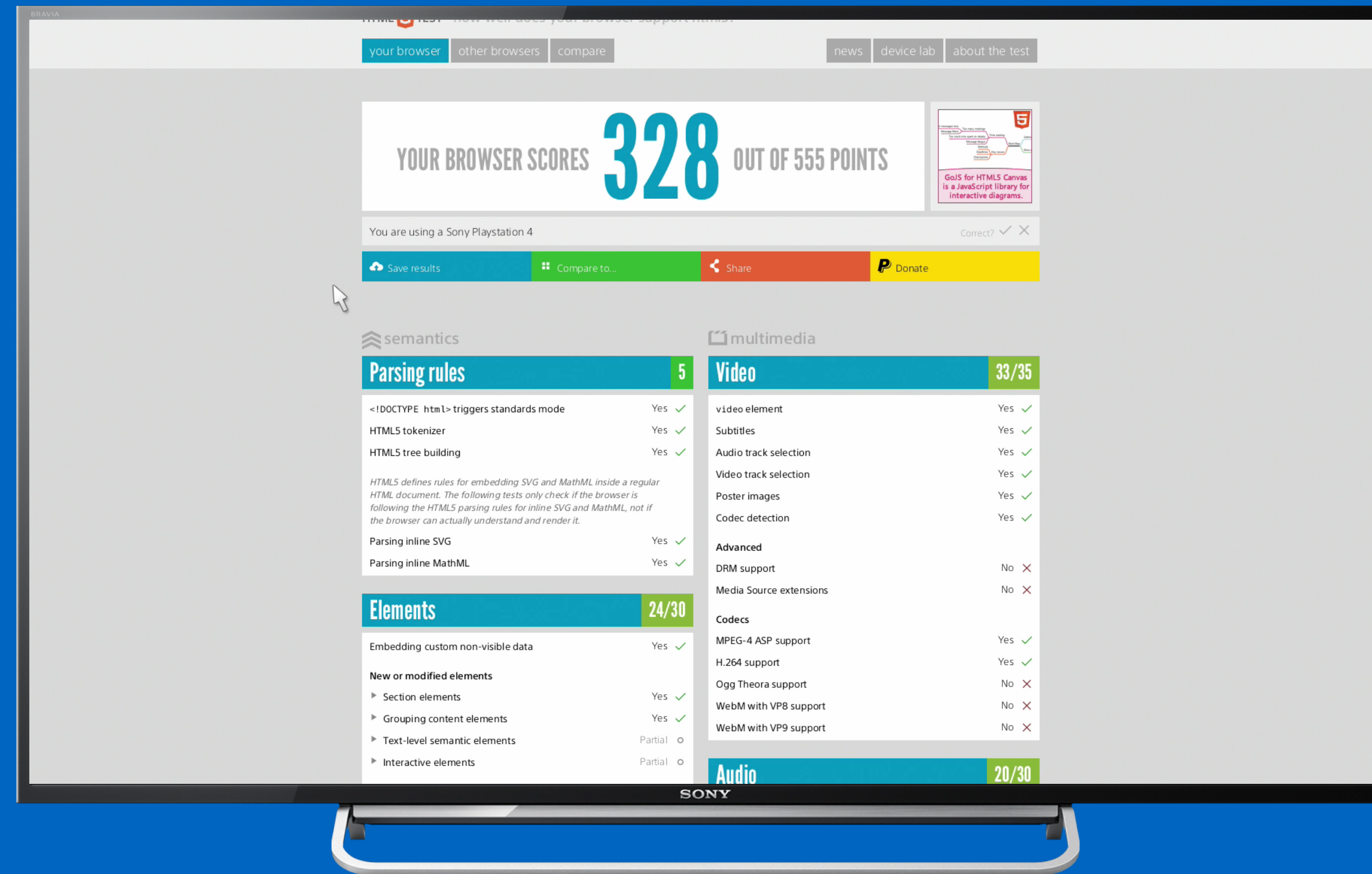
**configure the correct overscan correction  
in the system preferences**





the playstation 4 will always show the browser without overscan correction in full screen mode





the playstation 4 will always show the browser without overscan correction in full screen mode



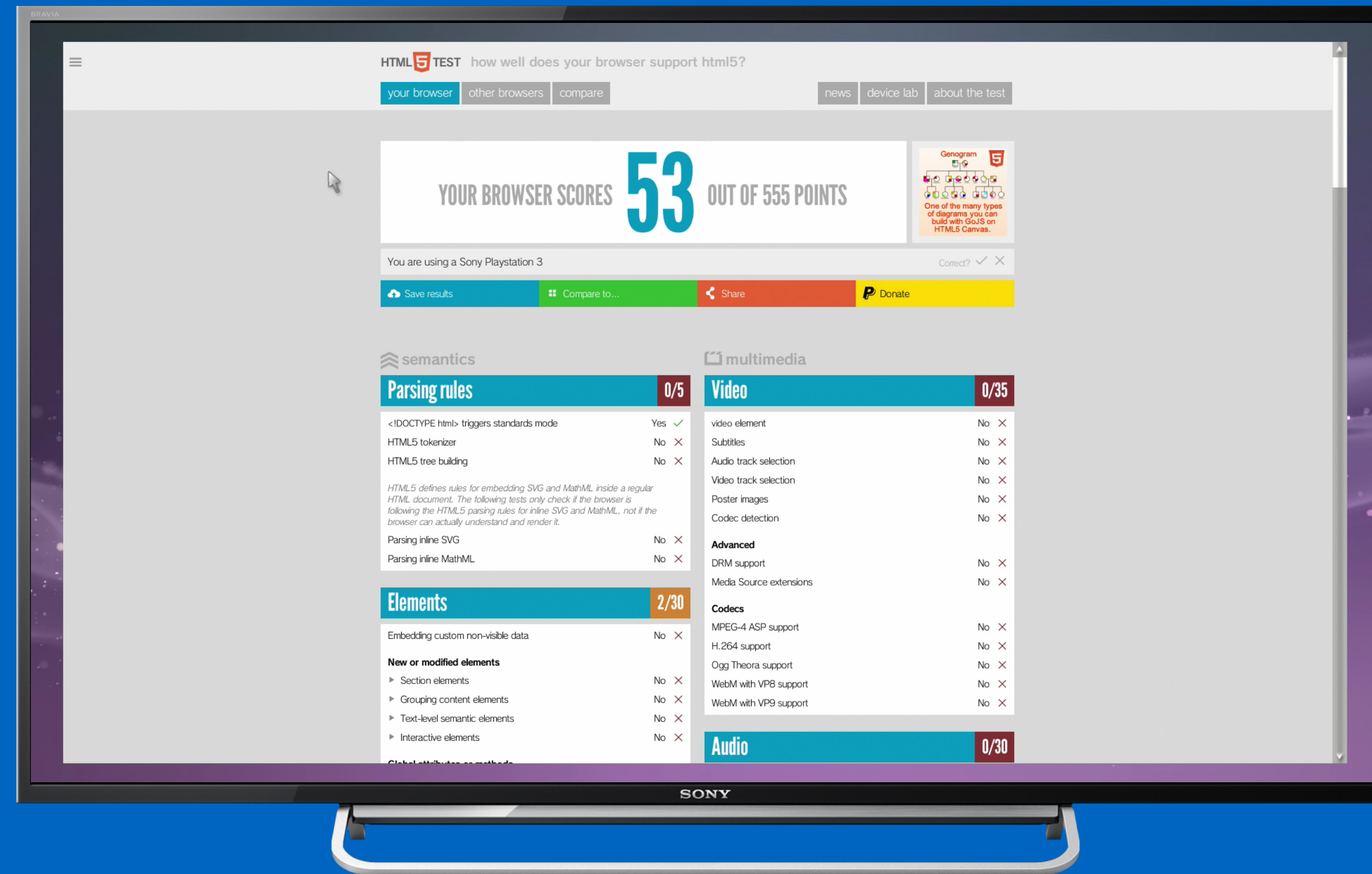
**solution 2**  
**no overscan**



**it is possible to disable overscan  
on many television sets**

**‘screen fit’, ‘pixel perfect’ or ‘just scan’**





the playstation 3 always shows the  
browser with overscan correction



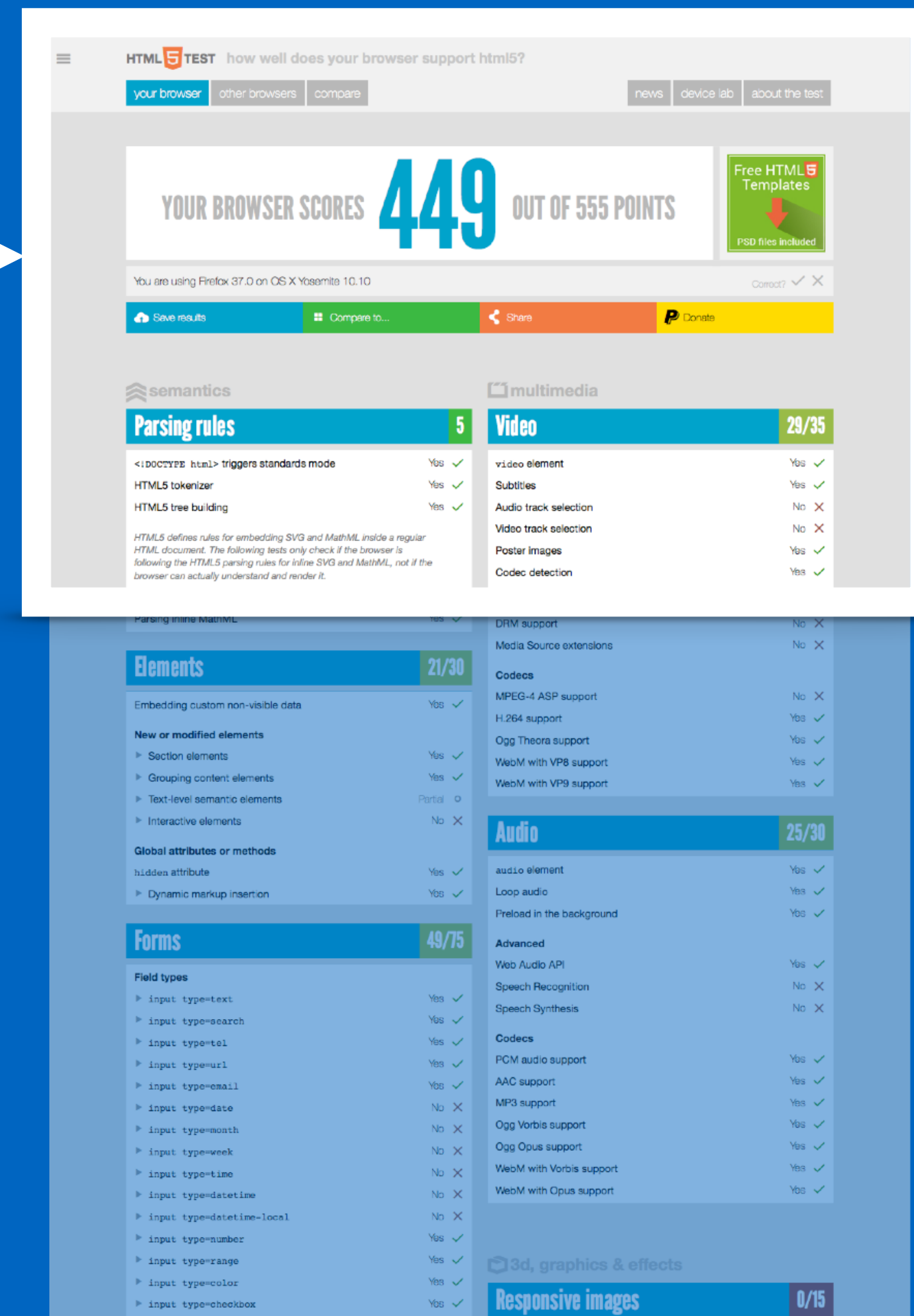
the viewport  
(i really need some aspirin!)



# the visual viewport

the visual viewport  
determines which  
part of the website  
will be visible

measured in  
device pixels





# the visual viewport

the visual viewport  
determines which  
part of the website  
will be visible

measured in  
device pixels

Clipboard	Yes	✓
Clipboard API and events	Yes	✓
Spellcheck	Yes	✓
spellcheck attribute	Yes	✓
Performance		25
Native binary data	Yes	✓
Workers		
Web Workers	Yes	✓
Shared Workers	Yes	✓
Security		33/41
Web Cryptography API	Yes	✓
Content Security Policy 1.0	Yes	✓
Content Security Policy 1.1	No	✗
Cross-Origin Resource Sharing	Yes	✓
Cross-document messaging	Yes	✓
Iframes		
Sandboxed iframe	Yes	✓
Seamless iframe	No	✗
Iframe with inline contents	Yes	✓
History and navigation		10
Session history	Yes	✓
Web Components		2/10
Custom elements	No	✗
Shadow DOM	No	✗
HTML templates	Yes	✓
HTML imports	No	✗
Other		20
Styling		
Scoped style element	Yes	✓
Scripting	Yes	✓
Asynchronous scripting	Yes	✓
Runtime error reporting	Yes	✓
Script execution events	Yes	✓
Base64 encoding and decoding	Yes	✓
JSON encoding and decoding	Yes	✓
URL API	Yes	✓
Mutation Observer	Yes	✓
Promises	Yes	✓
Other		
Page Visibility	Yes	✓
Text selection	Yes	✓
Scroll into view	Yes	✓





# the visual viewport

the visual viewport  
determines which  
part of the website  
will be visible

measured in  
device pixels

HTML5 TEST how well does your browser support html5?

your browser other browsers compare news device lab about the test

YOUR BROWSER SCORES **449** OUT OF 555 POINTS

You are using Firefox 37.0 on OS X Yosemite 10.10

Save results Compare to... Share Donate

**semantics**

**Parsing rules** 5

<!DOCTYPE html> triggers standards mode	Yes	✓
HTML5 tokenizer	Yes	✓
HTML5 tree building	Yes	✓

HTML5 defines rules for embedding SVG and MathML inside a regular HTML document. The following tests only check if the browser is following the HTML5 parsing rules for inline SVG and MathML, not if the browser can actually understand and render it.

**multimedia**

**Video** 29/35

video element	Yes	✓
Subtitles	Yes	✓
Audio track selection	No	✗
Video track selection	No	✗
Poster images	Yes	✓
Codec detection	Yes	✓

**Elements** 21/30

Embedding custom non-visible data	Yes	✓
New or modified elements		
> Section elements	Yes	✓
> Grouping content elements	Yes	✓
> Text-level semantic elements	Partial	○
> Interactive elements	No	✗
Global attributes or methods		
hidden attribute	Yes	✓
Dynamic markup insertion	Yes	✓

**Forms** 49/75

Field types		
> input type=text	Yes	✓
> input type=search	Yes	✓
> input type=tel	Yes	✓
> input type=url	Yes	✓
> input type=email	Yes	✓
> input type=date	No	✗
> input type=month	No	✗
> input type=week	No	✗
> input type=time	No	✗
> input type=datetime	No	✗
> input type=datetime-local	No	✗
> input type=number	Yes	✓
> input type=range	Yes	✓
> input type=color	Yes	✓
> input type=checkbox	Yes	✓

**Audio** 25/30

audio element	Yes	✓
Loop audio	Yes	✓
Preload in the background	Yes	✓
Advanced		
Web Audio API	Yes	✓
Speech Recognition	No	✗
Speech Synthesis	No	✗
Codecs		
PCM audio support	Yes	✓
AAC support	Yes	✓
MP3 support	Yes	✓
Ogg Vorbis support	Yes	✓
Ogg Opus support	Yes	✓
WebM with Vorbis support	Yes	✓
WebM with Opus support	Yes	✓

**3d, graphics & effects**

**Responsive images** 0/15



# the layout viewport



the layout viewport  
determines the  
width in css pixels  
on which the site  
will be rendered

HTML5 TEST how well does your browser support html5?

your browser other browsers compare news device lab about the test

YOUR BROWSER SCORES **449** OUT OF 555 POINTS

You are using Firefox 37.0 on OS X Yosemite 10.10

Save results Compare to... Share Donate

**semantics**

**Parsing rules** 5

- <!DOCTYPE html> triggers standards mode Yes ✓
- HTML5 tokenizer Yes ✓
- HTML5 tree building Yes ✓
- HTML5 defines rules for embedding SVG and MathML inside a regular HTML document. The following tests only check if the browser is following the HTML5 parsing rules for inline SVG and MathML, not if the browser can actually understand and render it.
- Parsing inline SVG Yes ✓
- Parsing inline MathML Yes ✓

**Elements** 21/30

- Embedding custom non-visible data Yes ✓
- New or modified elements**
- Section elements Yes ✓
- Grouping content elements Yes ✓
- Text-level semantic elements Partial ○
- Interactive elements No ✗
- Global attributes or methods**
- hidden attribute Yes ✓
- Dynamic markup insertion Yes ✓

**Forms** 49/75

**Field types**

- input type=text Yes ✓
- input type=search Yes ✓
- input type=tel Yes ✓
- input type=url Yes ✓
- input type=email Yes ✓
- input type=date No ✗
- input type=month No ✗
- input type=week No ✗
- input type=time No ✗
- input type=datetime No ✗
- input type=datetime-local No ✗
- input type=number Yes ✓
- input type=range Yes ✓
- input type=color Yes ✓
- input type=checkbox Yes ✓

**multimedia**

**Video** 29/35

- video element Yes ✓
- Subtitles Yes ✓
- Audio track selection No ✗
- Video track selection No ✗
- Poster images Yes ✓
- Codec detection Yes ✓
- Advanced**
- DRM support No ✗
- Media Source extensions No ✗
- Codecs**
- MPEG-4 ASP support No ✗
- H.264 support Yes ✓
- Ogg Theora support Yes ✓
- WebM with VP8 support Yes ✓
- WebM with VP9 support Yes ✓

**Audio** 25/30

- audio element Yes ✓
- Loop audio Yes ✓
- Preload in the background Yes ✓
- Advanced**
- Web Audio API Yes ✓
- Speech Recognition No ✗
- Speech Synthesis No ✗
- Codecs**
- PCM audio support Yes ✓
- AAC support Yes ✓
- MP3 support Yes ✓
- Ogg Vorbis support Yes ✓
- Ogg Opus support Yes ✓
- WebM with Vorbis support Yes ✓
- WebM with Opus support Yes ✓

**3d, graphics & effects**

**Responsive images** 0/15



# the layout viewport



the layout viewport  
determines the  
width in css pixels  
on which the site  
will be rendered

HTML5 TEST how well does your browser support html5?

your browser other browsers compare news device lab about the test

YOUR BROWSER SCORES

449

OUT OF 555 POINTS

You are using Firefox 37.0 on OS X Yosemite 10.10

Save results Compare to... Share Donate

semantics

Parsing rules 5

<DOCTYPE html> triggers standards mode	Yes ✓
HTML5 tokenizer	Yes ✓
HTML5 tree building	Yes ✓
HTML5 defines rules for embedding SVG and MathML inside a regular HTML document. The following tests only check if the browser is following the HTML5 parsing rules for inline SVG and MathML, not if the browser can actually understand and render it.	
Parsing inline SVG	Yes ✓
Parsing inline MathML	Yes ✓

Elements 21/30

Embedding custom non-visible data	Yes ✓
New or modified elements	
Section elements	Yes ✓
Grouping content elements	Yes ✓
Text-level semantic elements	Partial ◐
Interactive elements	No ✗
Global attributes or methods	
hidden attribute	Yes ✓
Dynamic markup insertion	Yes ✓

Forms 49/75

Field types	
input type=text	Yes ✓
input type=search	Yes ✓
input type=tel	Yes ✓
input type=url	Yes ✓
input type=email	Yes ✓
input type=date	No ✗
input type=month	No ✗
input type=week	No ✗
input type=time	No ✗
input type=datetime	No ✗

multimedia

Video 29/35

video element	Yes ✓
Subtitles	Yes ✓
Audio track selection	No ✗
Video track selection	No ✗
Poster images	Yes ✓
Codec detection	Yes ✓
Advanced	
DRM support	No ✗
Media Source extensions	No ✗
Codecs	
MPEG-4 ASP support	No ✗
H.264 support	Yes ✓
Ogg Theora support	Yes ✓
WebM with VP8 support	Yes ✓
WebM with VP9 support	Yes ✓

Audio 25/30

audio element	Yes ✓
Loop audio	Yes ✓
Preload in the background	Yes ✓
Advanced	
Web Audio API	Yes ✓
Speech Recognition	No ✗
Speech Synthesis	No ✗
Codecs	
PCM audio support	Yes ✓
AAC support	Yes ✓
MP3 support	Yes ✓
Ogg Vorbis support	Yes ✓
Ogg Opus support	Yes ✓
WebM with Vorbis support	Yes ✓
WebM with Opus support	Yes ✓



# the layout viewport



the layout viewport  
determines the  
width in css pixels  
on which the site  
will be rendered



**the default layout viewport is different on  
every smart tv, console or set-top box**

**between 800 and 1920 css pixels**



it is possible to change the width of the layout viewport with the 'meta viewport' tag

$$\frac{\text{physical device pixels}}{\text{device scale factor}}$$



```
<meta name="viewport" content="width=device-width">
```

```
<meta name="viewport" content="width=1024">
```



**complication:**

**meta viewport is not supported**

**it is not possible to get the same layout viewport  
width in all of the different browsers**



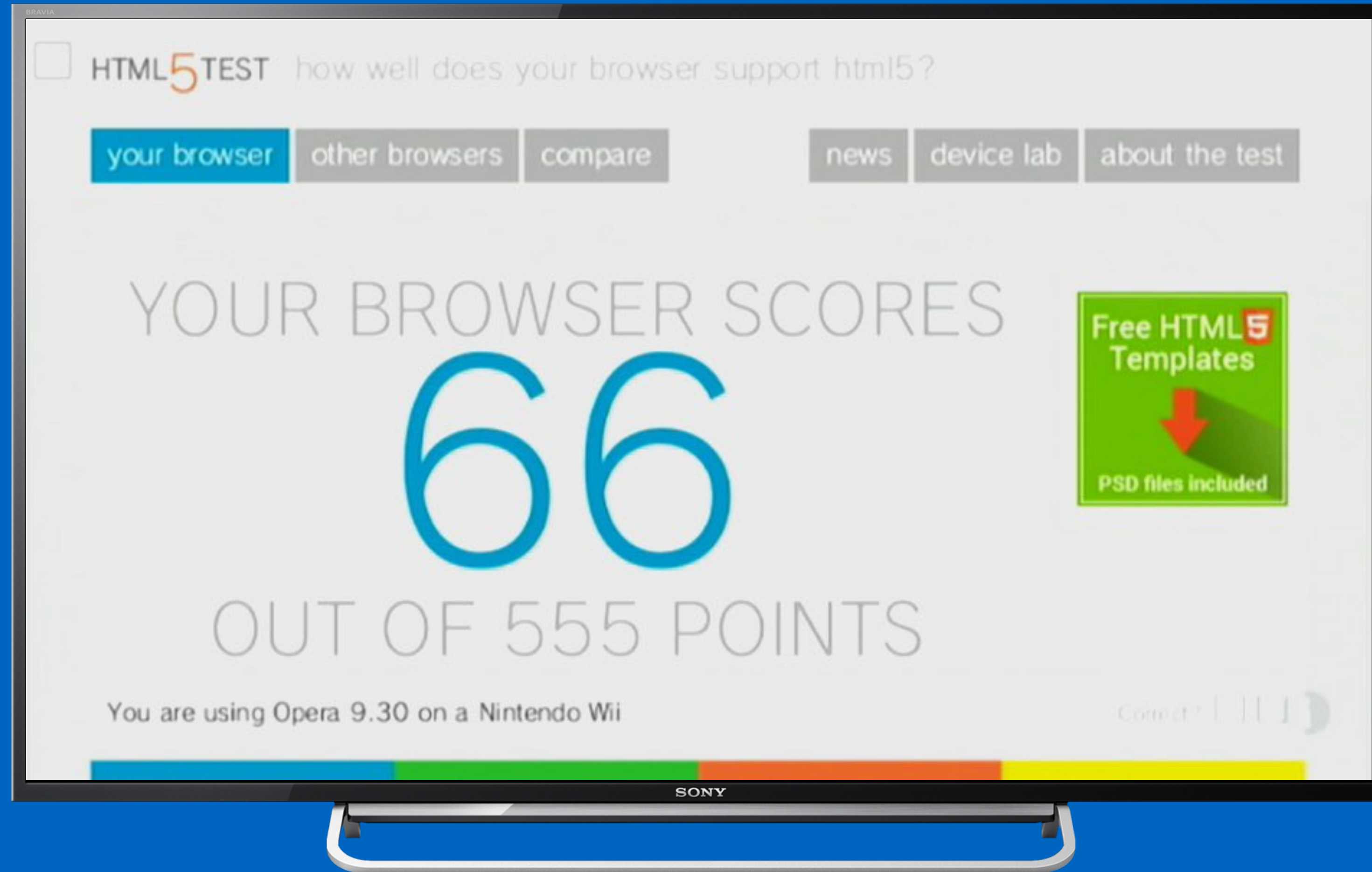
**complication:**

**device pixel ratio is not supported**

**there is no proper way to show images with the same  
resolution as the physical screen**



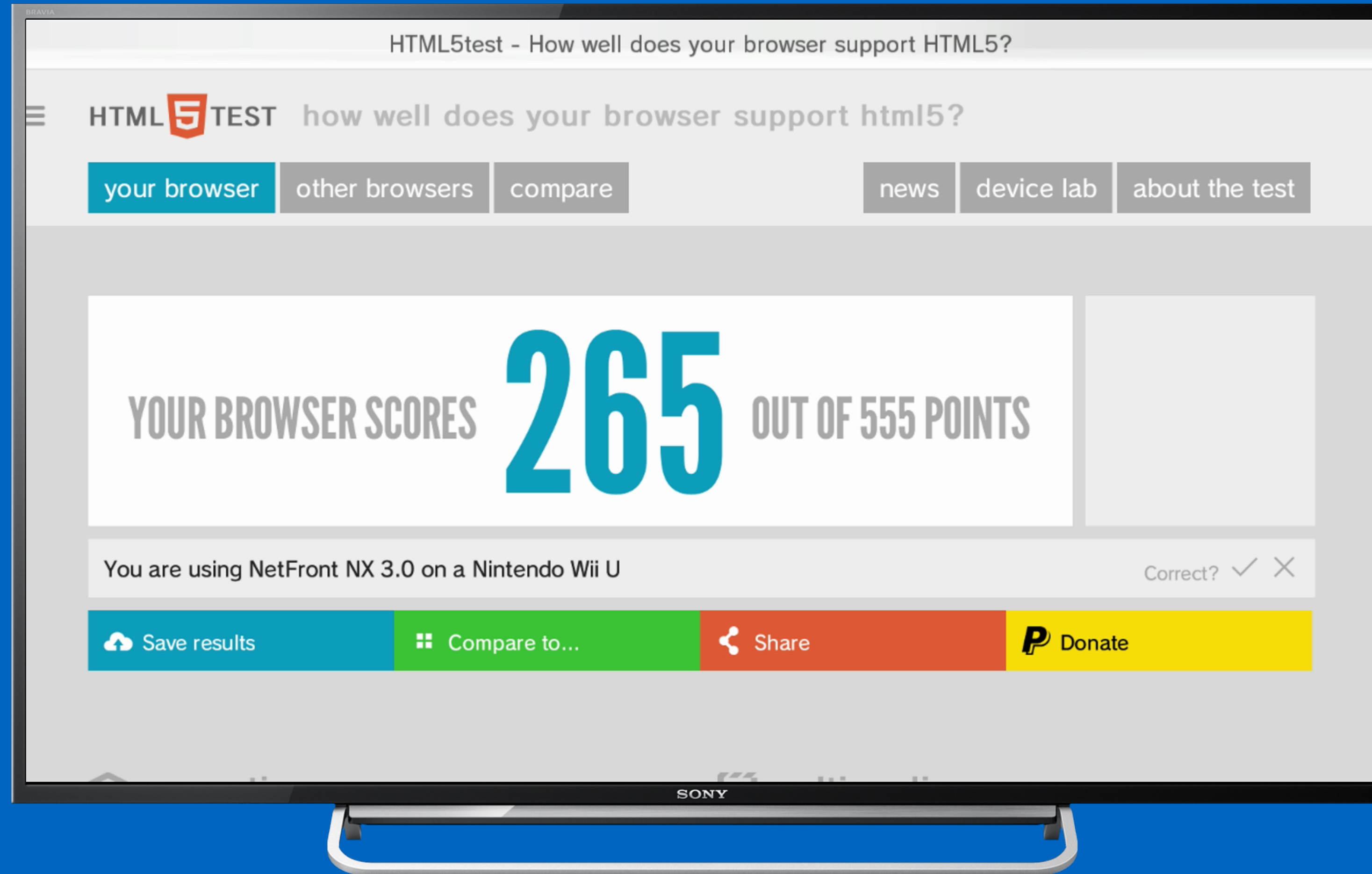
800 pixels



nintendo wii



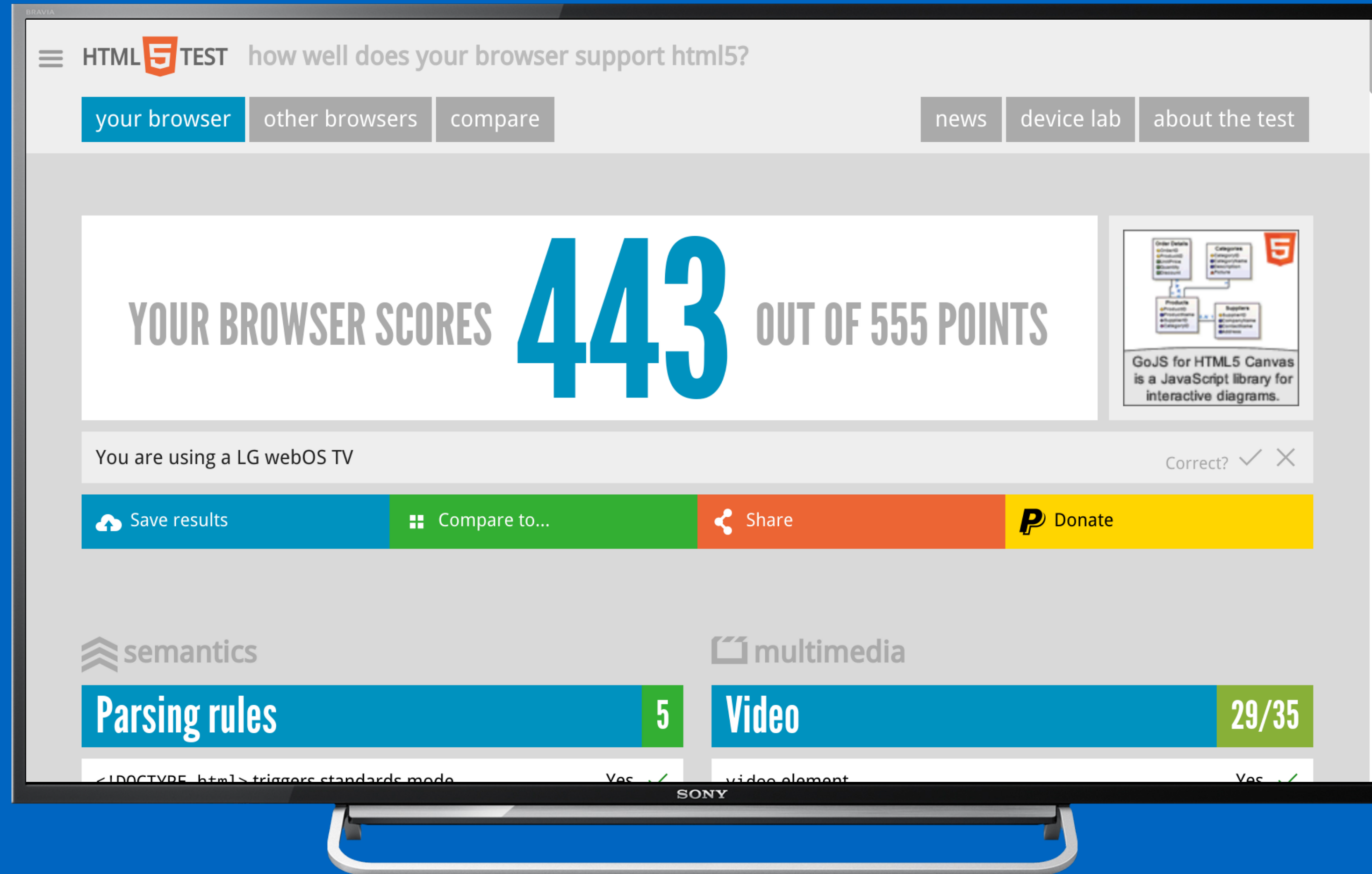
980 pixels



nintendo wii u



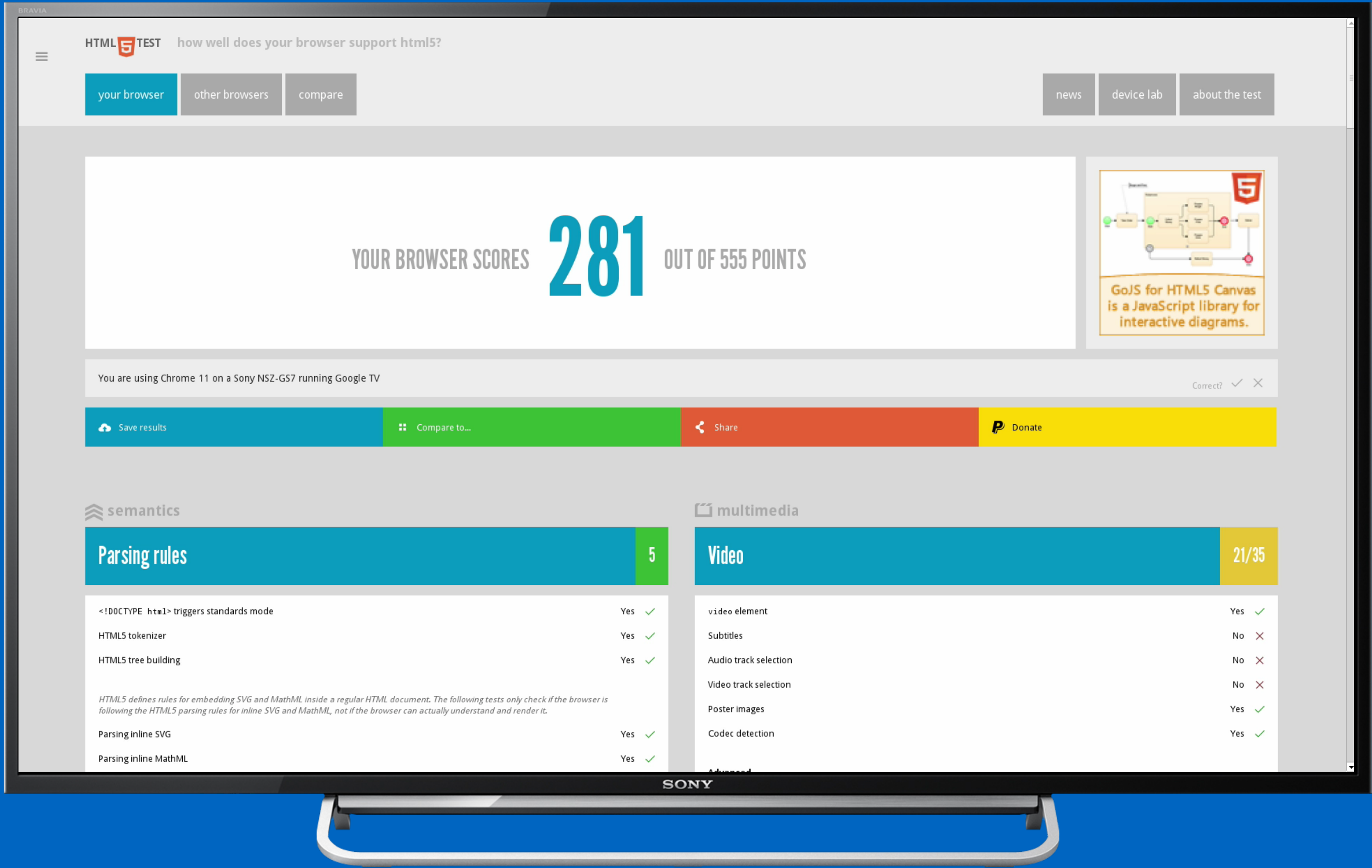
960 pixels



lg webos



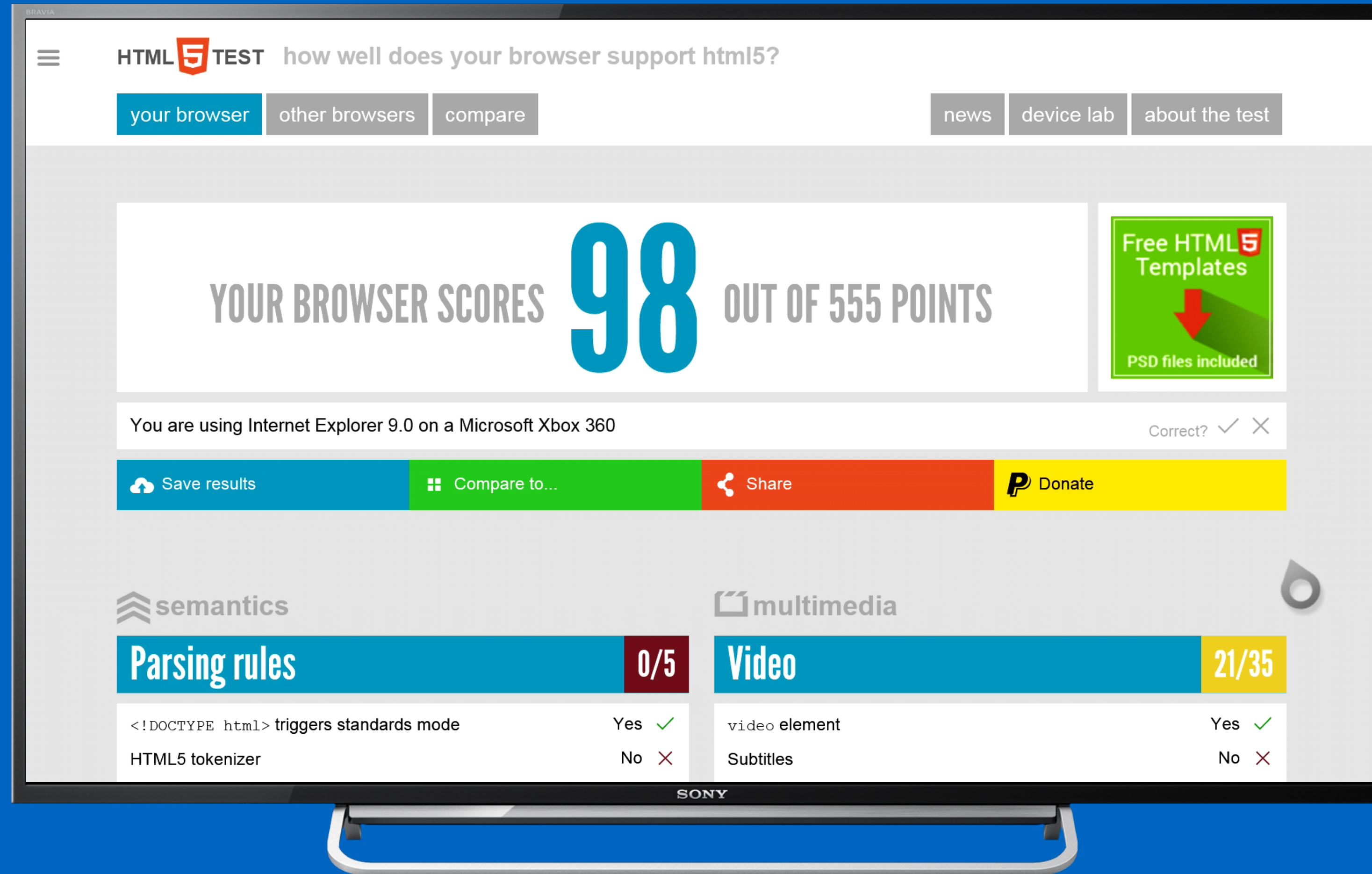
1024 pixels



google tv



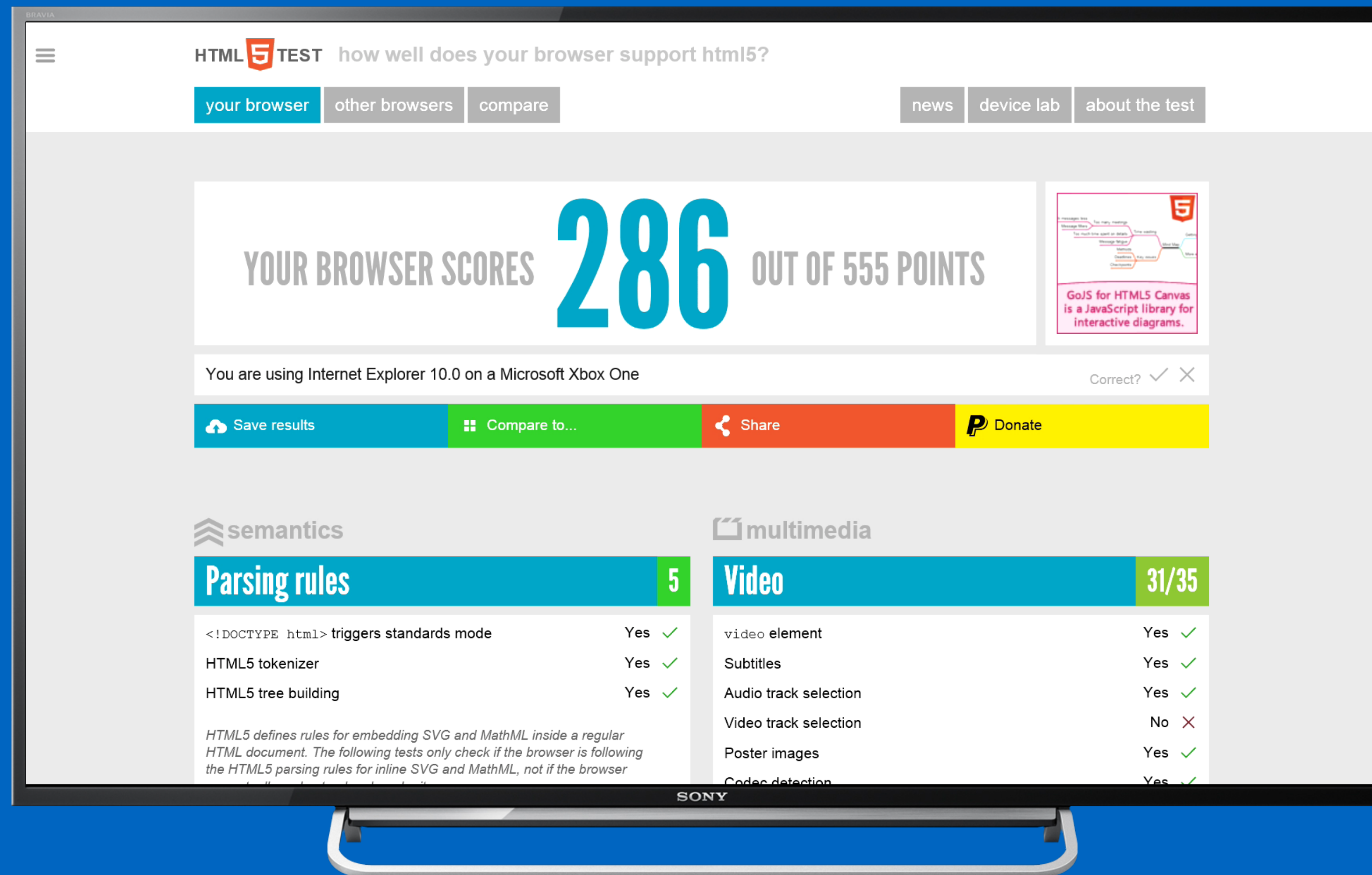
1041 of 1050 pixels



microsoft xbox 360



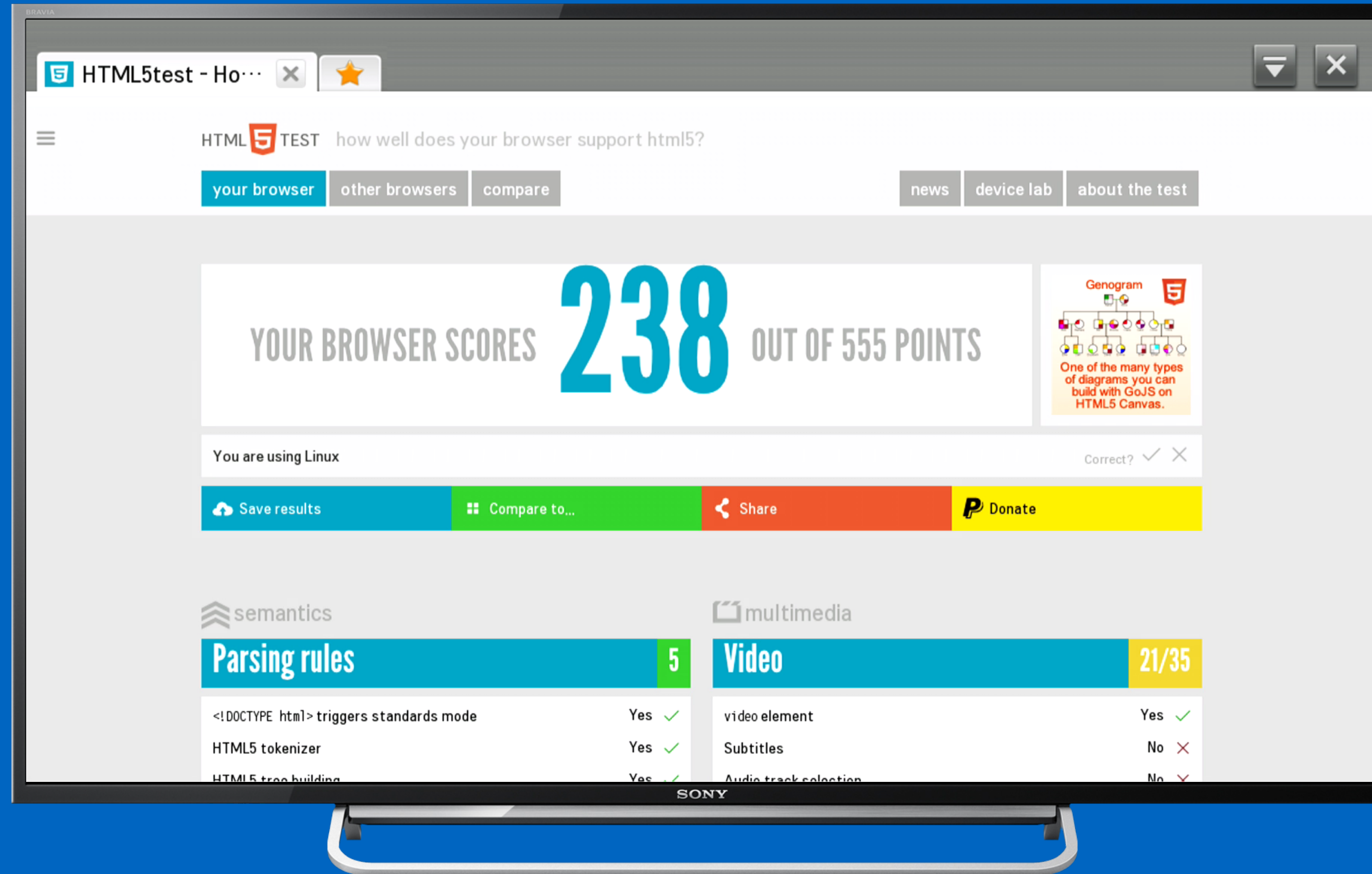
1200 of 1236 pixels



microsoft xbox one



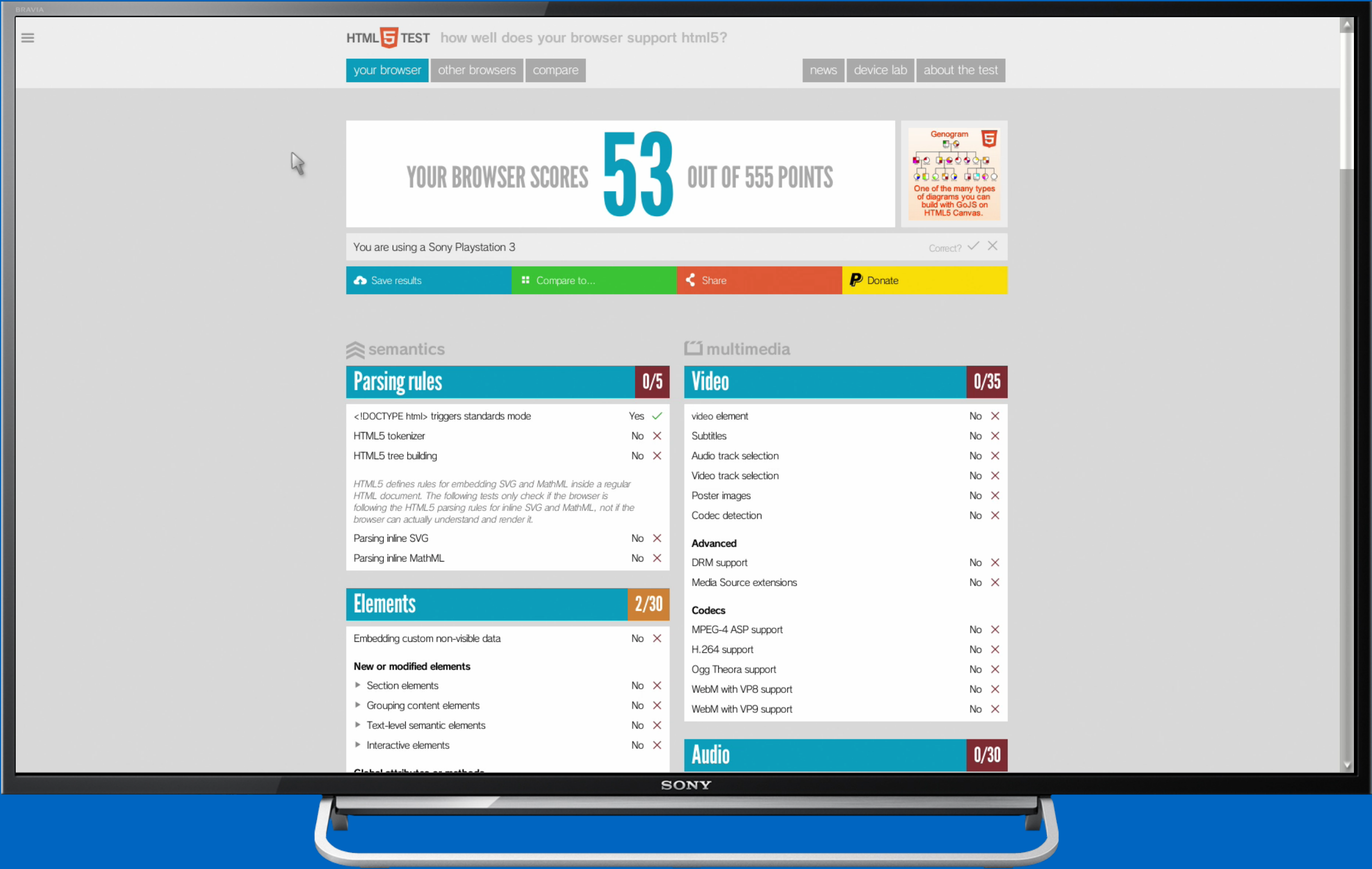
1226 pixels



lg netcast



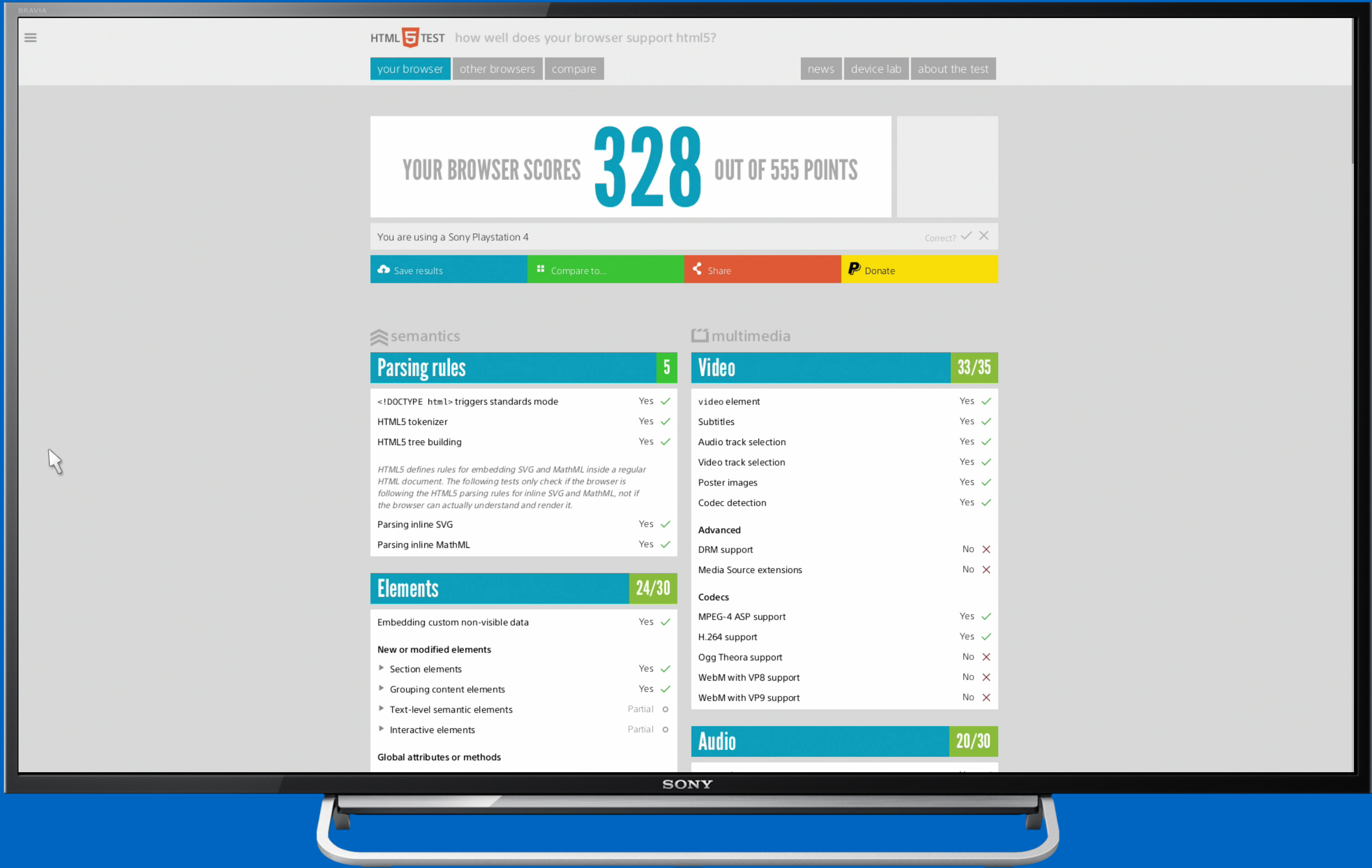
1824 pixels



sony playstation 3



1920 pixels



sony playstation 4







device pixels  $\neq$  device pixels  
(of course not)



**sometimes devices pixels are not  
physical devices pixels, but virtual device pixels**

**the browser renders in a lower resolution  
which is upscaled to the resolution of the display**



3

distance to the screen



“Make fonts and graphics on the site larger to account for viewing distance. People sit proportionally farther from a TV than from a computer monitor of the same size.”

– **Internet Explorer for Xbox One Developer Guide**



# **fluid design++**

**the size of the contents is determined  
by the width of the viewport**



# 1 use percentages for positioning

```
.left { width: 60%; }
```

```
.right { left: 60%; width: 40%; }
```



2

**base the fontsize on the viewport**

```
document.body.style.fontSize =  
    ((window.innerWidth / 1920) * 300) + '%';
```



### 3 or maybe use viewport units – with polyfill

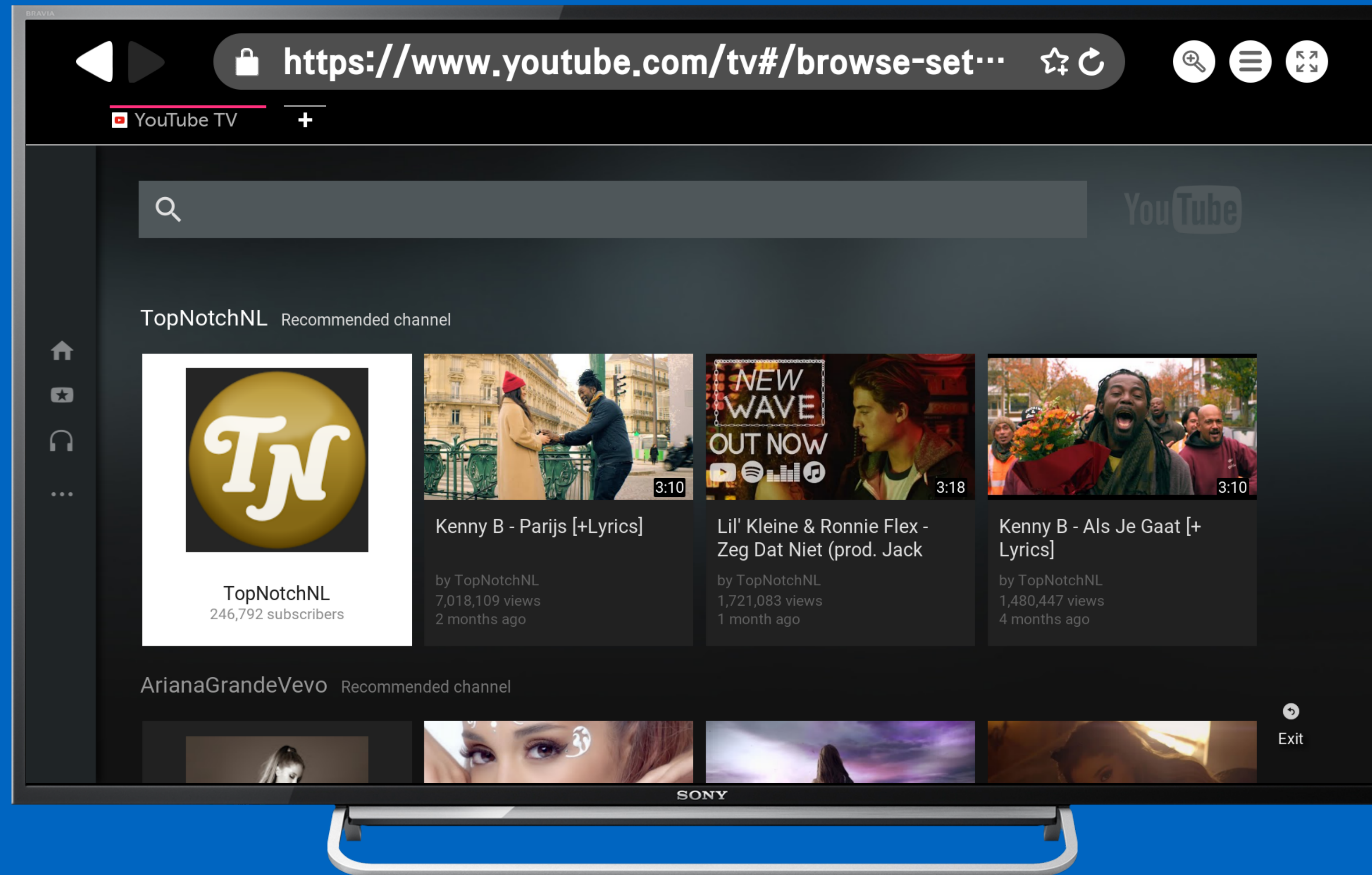
```
body { font-size: 3vw; }  
.left { width: 60vw; height: 100vh; }  
.right { width: 40vw; height: 100vh; }
```



## 4 use a safe margin around the contents

```
body {  
    padding: 5%;  
}
```





youtube tv website



# identifying smart tv's

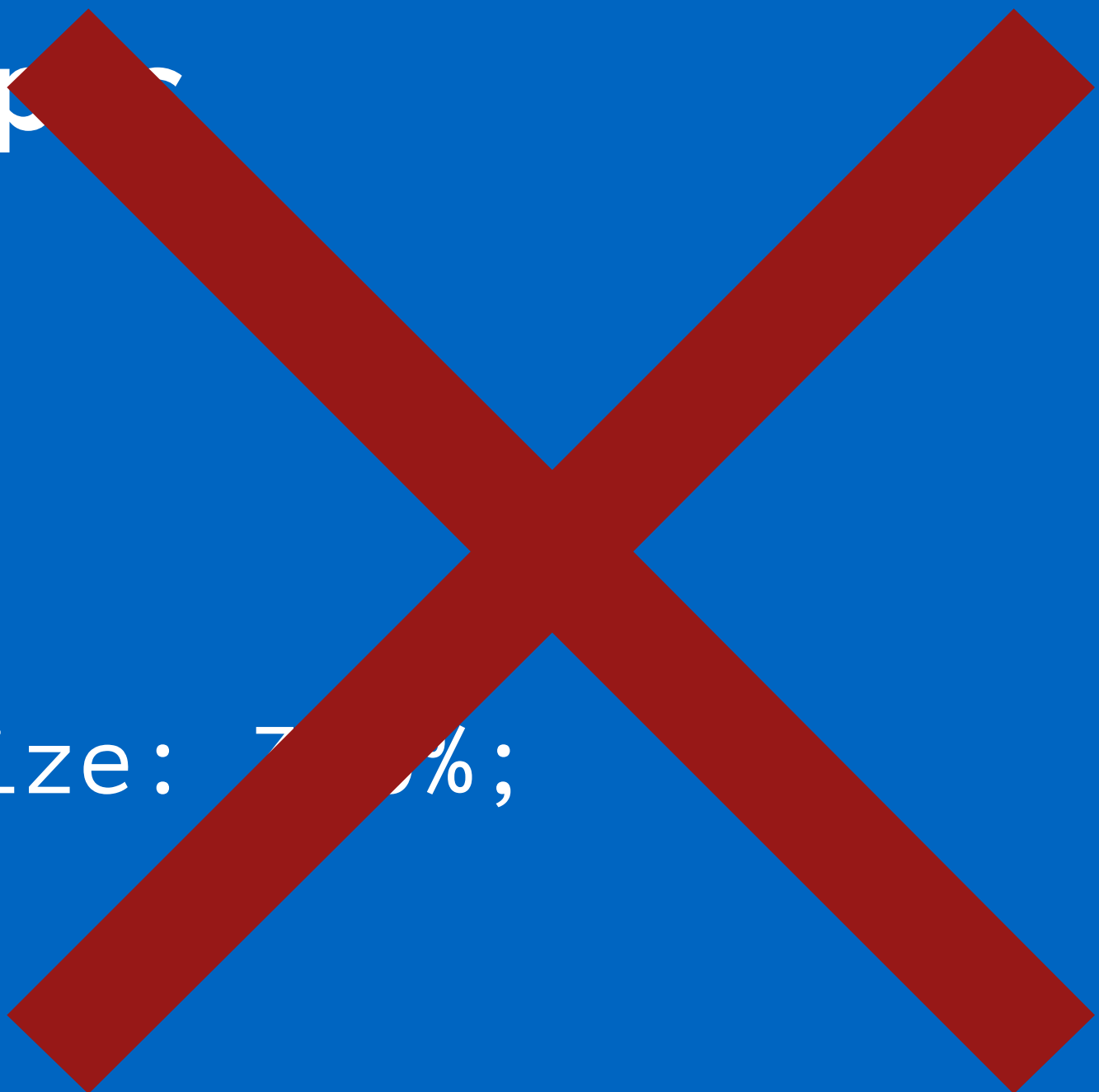
(css for televisions)



1

## css media types

```
@media tv {  
  body {  
    font-size: 70%;  
  }  
}
```





# 1 css media types

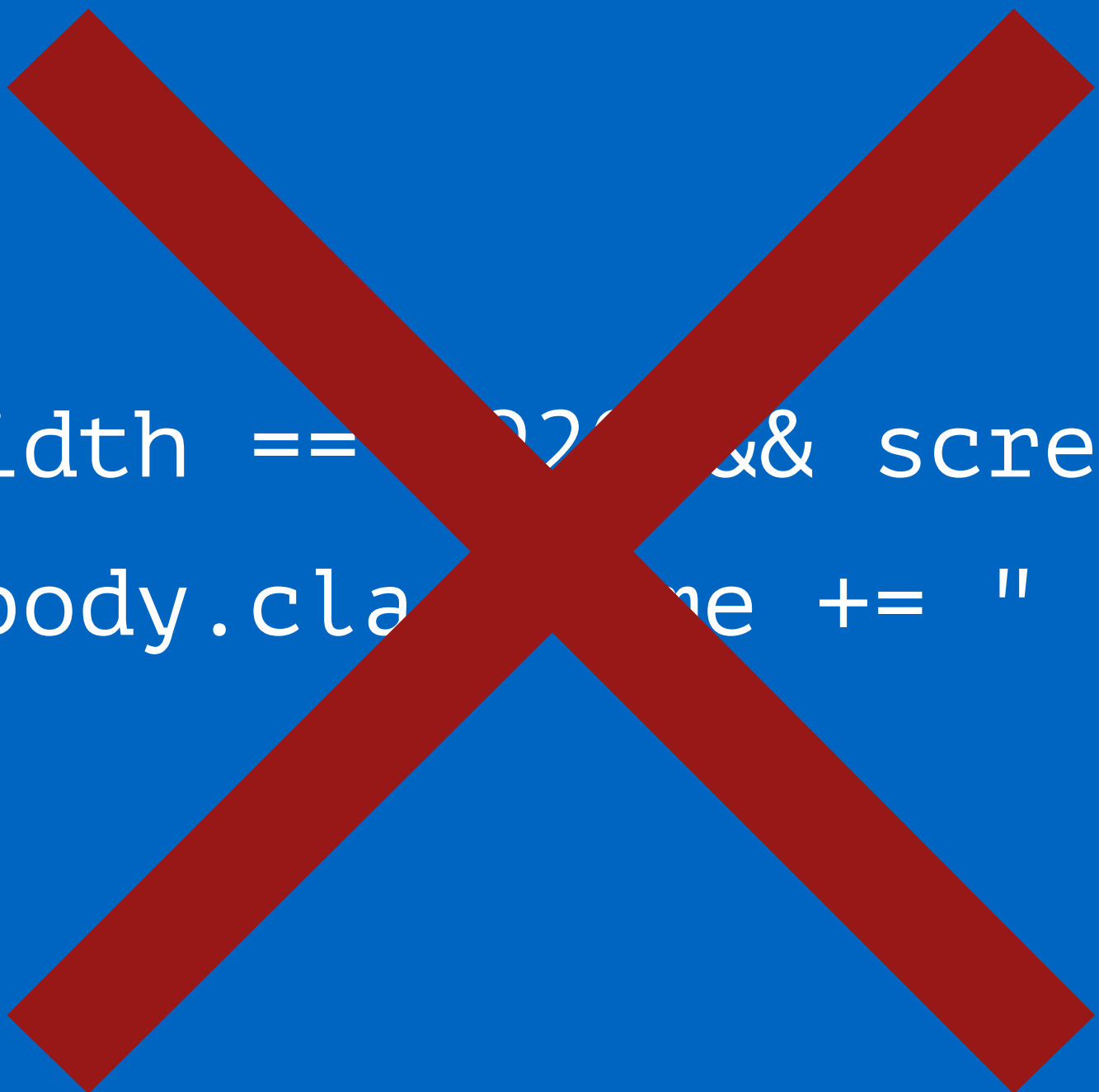
**all television browsers use the  
css media type 'screen'**



2

## screen size

```
if (screen.width == 1280 & screen.height == 1080) {  
    document.body.className += " television";  
}
```





2

screen size

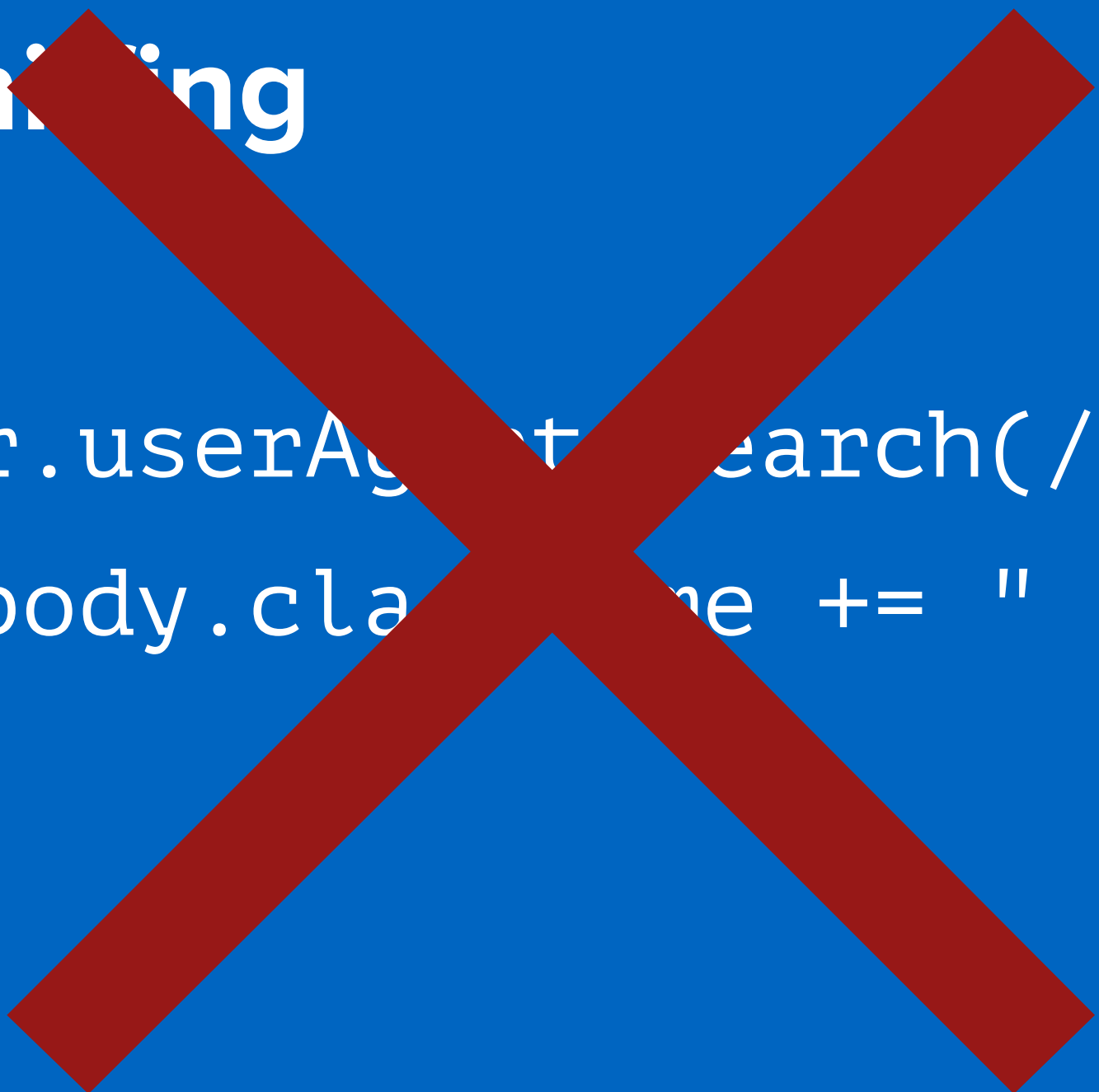
**monitors and phones often use  
hd resolutions, television browsers  
often use other resolutions**



3

## useragent sniffing

```
if (navigator.userAgent.search(/TV/i) >= 0) {  
    document.body.className += " television";  
}
```





### 3 useragent sniffing

**not all smart tv's are recognisable**

Mozilla/5.0 (X11; Linux; ko-KR)

AppleWebKit/534.26+ (KHTML, like Gecko)

Version/5.0 Safari/534.26+



## **4 couch mode**

**the only reliable way to optimise a website  
for television is to make two different websites...**

**or give the user the ability to switch on  
couch mode**



4

be careful with  
feature detection



“Basically every feature that talks to the operating system or hardware, is suspect.”

– Me



```
if (!!navigator.geolocation) {  
    navigator.geolocation.getCurrentPosition(  
        success, failure  
    );  
}  
else {  
    // alternative  
}
```



```
if (!!navigator.geolocation) {  
    navigator.geolocation.getCurrentPosition(  
        success, failure  
    );  
}
```

- 1 failure is called with a “permission denied” error code
- 2 no callback at all to success or failure



```
if (!!navigator.geolocation) {  
    navigator.geolocation.getCurrentPosition(  
        success, failure  
    );  
}
```

3

**success is called with longitude = 0 and latitude = 0**

4

**success is called with the coordinates of  
Mountain View, USA**



**is there a future for web apps  
on the big screen?**







**the new apple tv does not ship  
with a browser by default**



**android tv does not ship  
with a browser by default**

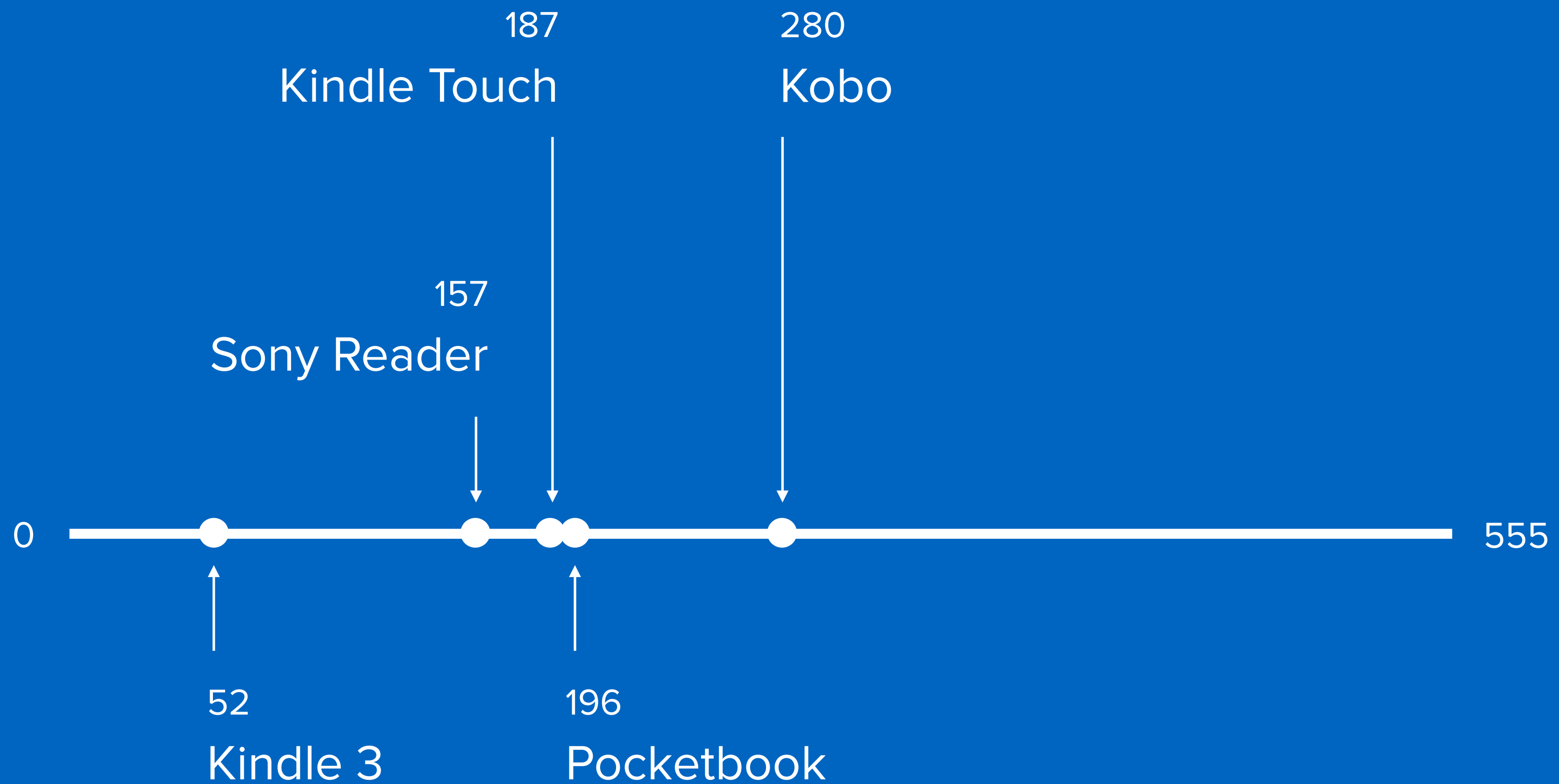






e-readers





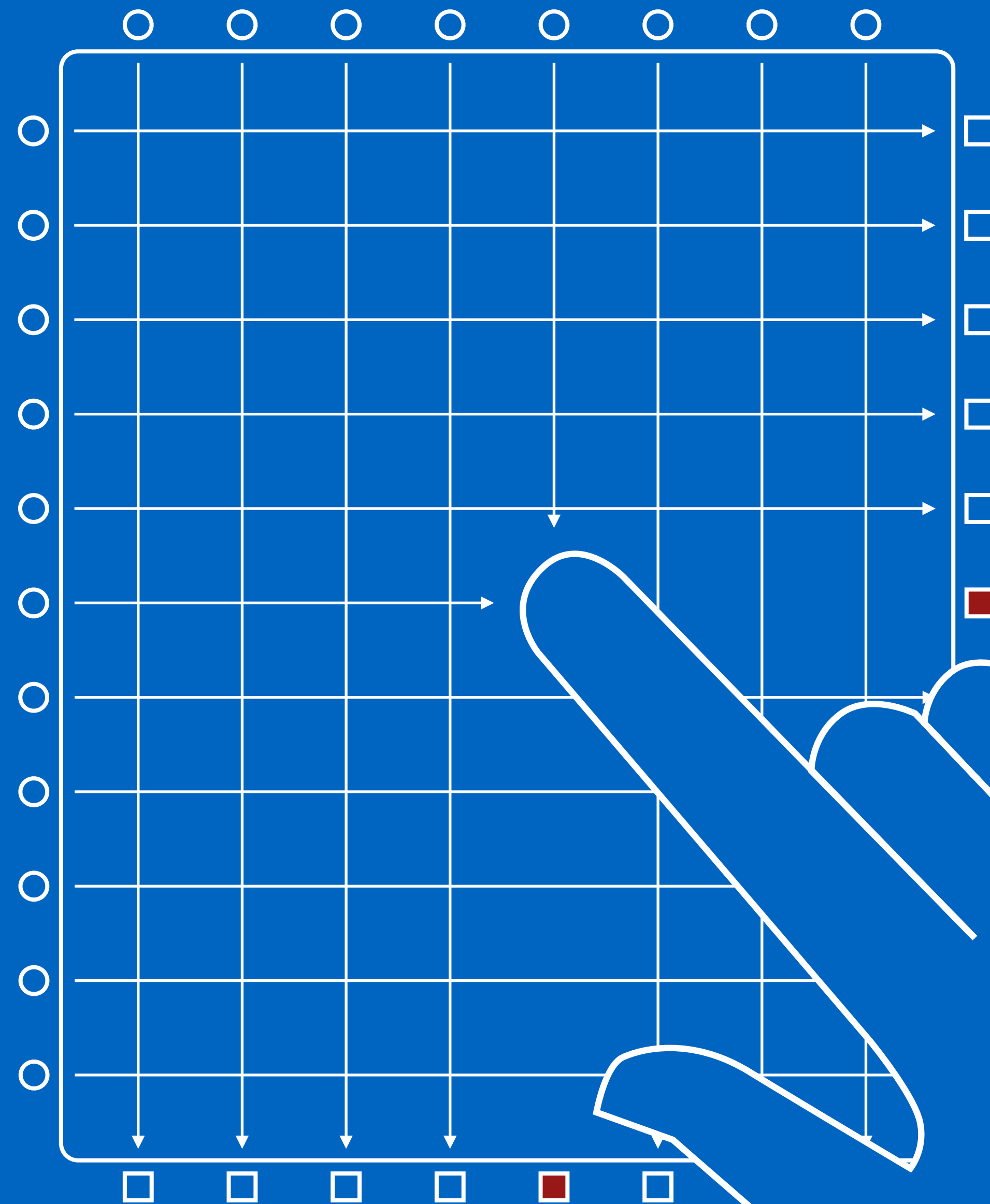
e-reader results on [html5test.com](http://html5test.com)



infrared touch screen



led's



sensors



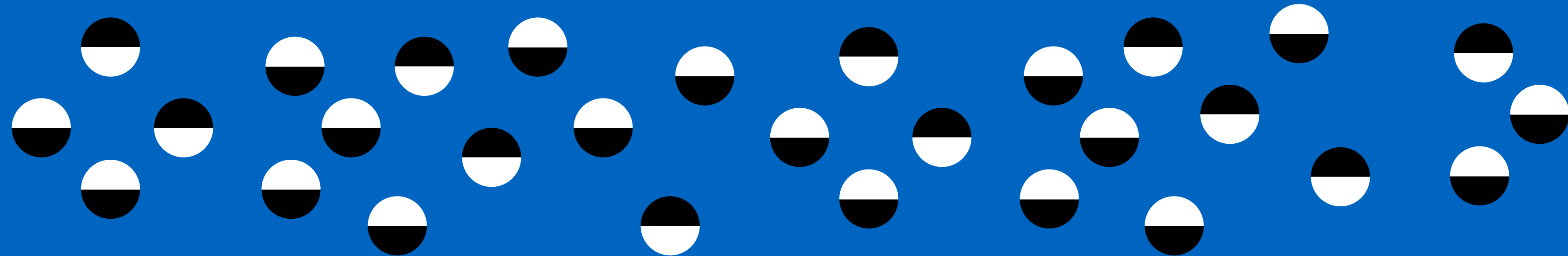
	mouse events		touch events
	down/up	move	
amazon kindle touch	yes		
pocketbook basic touch	yes		
kobo glow	yes	yes	
sony reader	yes	yes	1 finger



e-ink screens  
(slow, slower, slowest)



# microscopic electrostatic charged balls

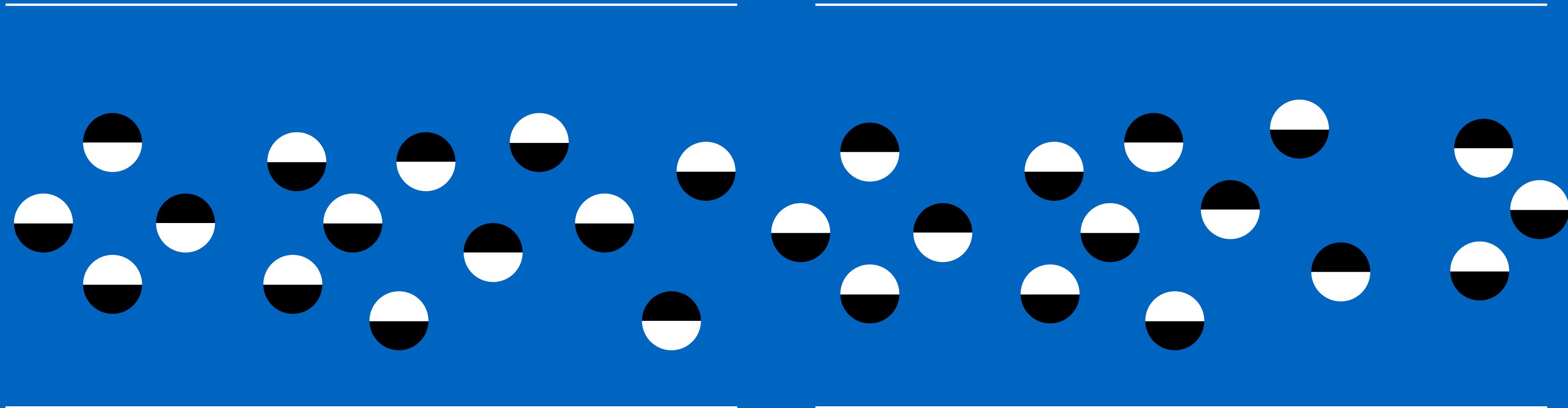




# microscopic electrostatic charged balls

+

-



-

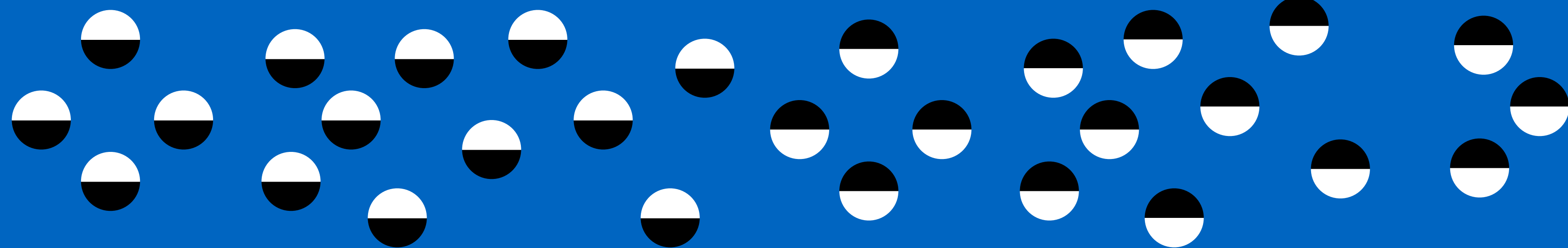
+



# microscopic electrostatic charged balls

+

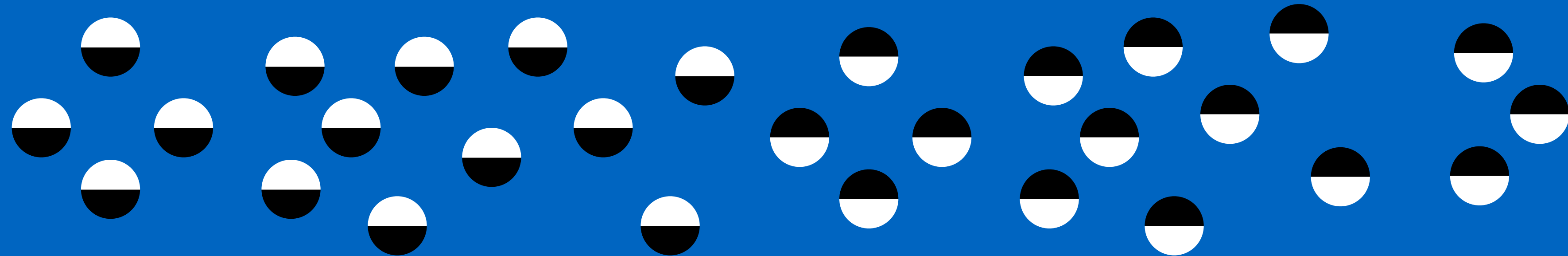
-



-

+

# microscopic electrostatic charged balls







HTML5test - How well do...

http://html5test.com/

HTML5 TEST

how well does your browser support html5?

your browser

other browsers

compare

news

device lab

about the test

YOUR BROWSER SCORES  
280

OUT OF 555 POINTS

kobo



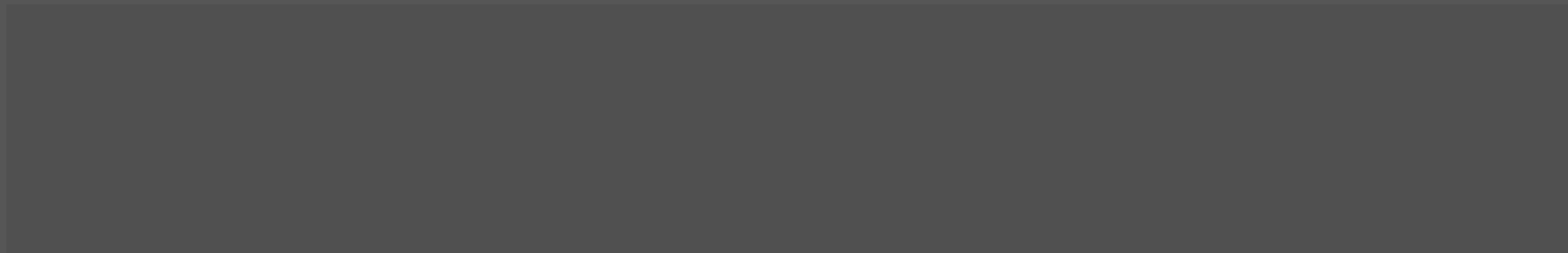
**maybe css animations and transitions  
weren't such a great idea after all**



**two completely different colors can look  
exactly the same in black and white**



**two completely different colors can look  
exactly the same in black and white**





# identifying e-readers

(css for e-ink screens)

1

css monochrome mediaquery

```
@media (monochrome)
```

```
...
```

```
}
```



# **1** **css monochrome mediaquery**

**all tested e-readers act like  
they have a color screen**

## **2 useragent sniffing**

**there is no universal marker in the useragent string, but we can recognise individual manufacturers and models**

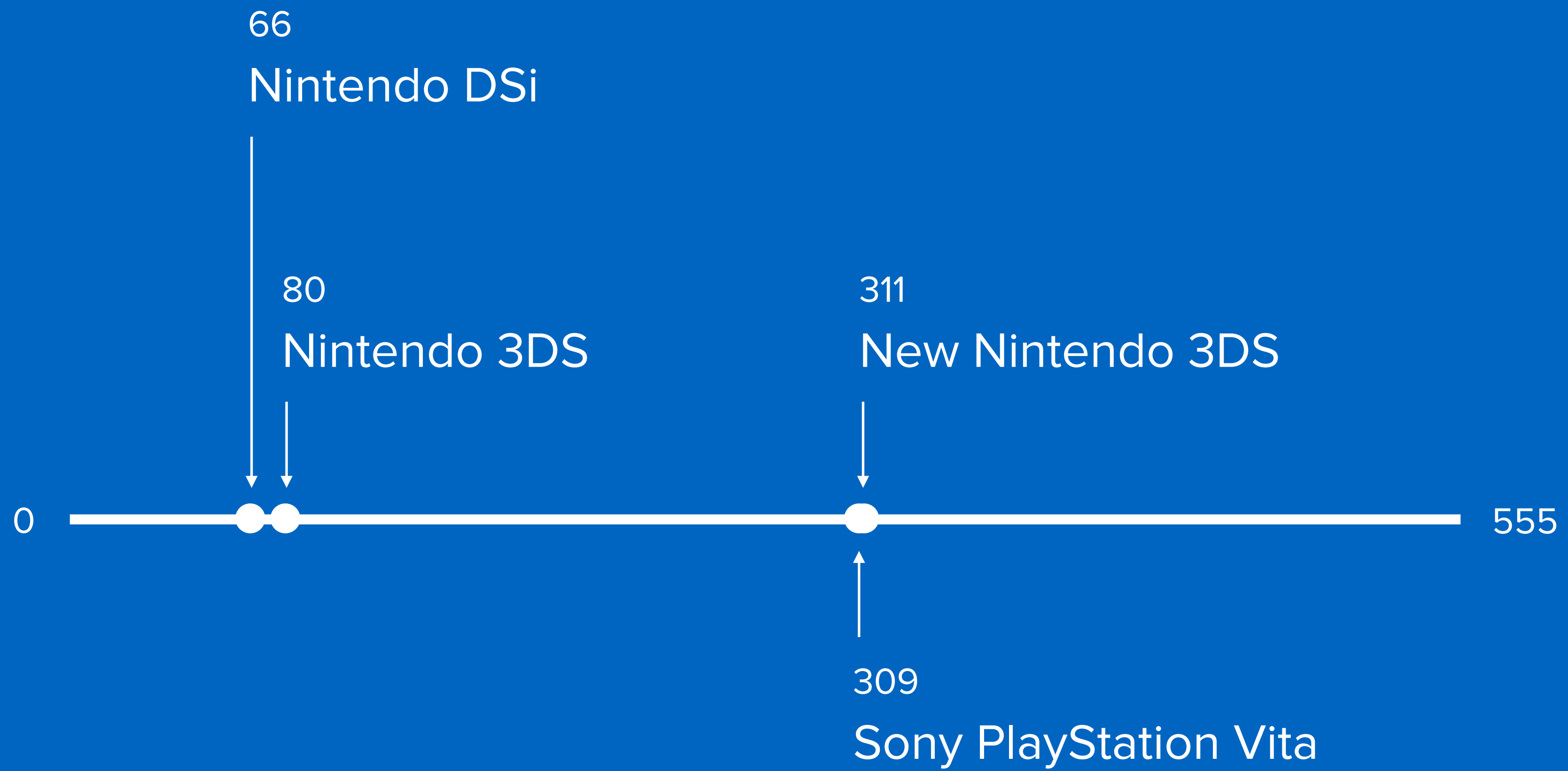






portable consoles





portable console results [html5test.com](http://html5test.com)





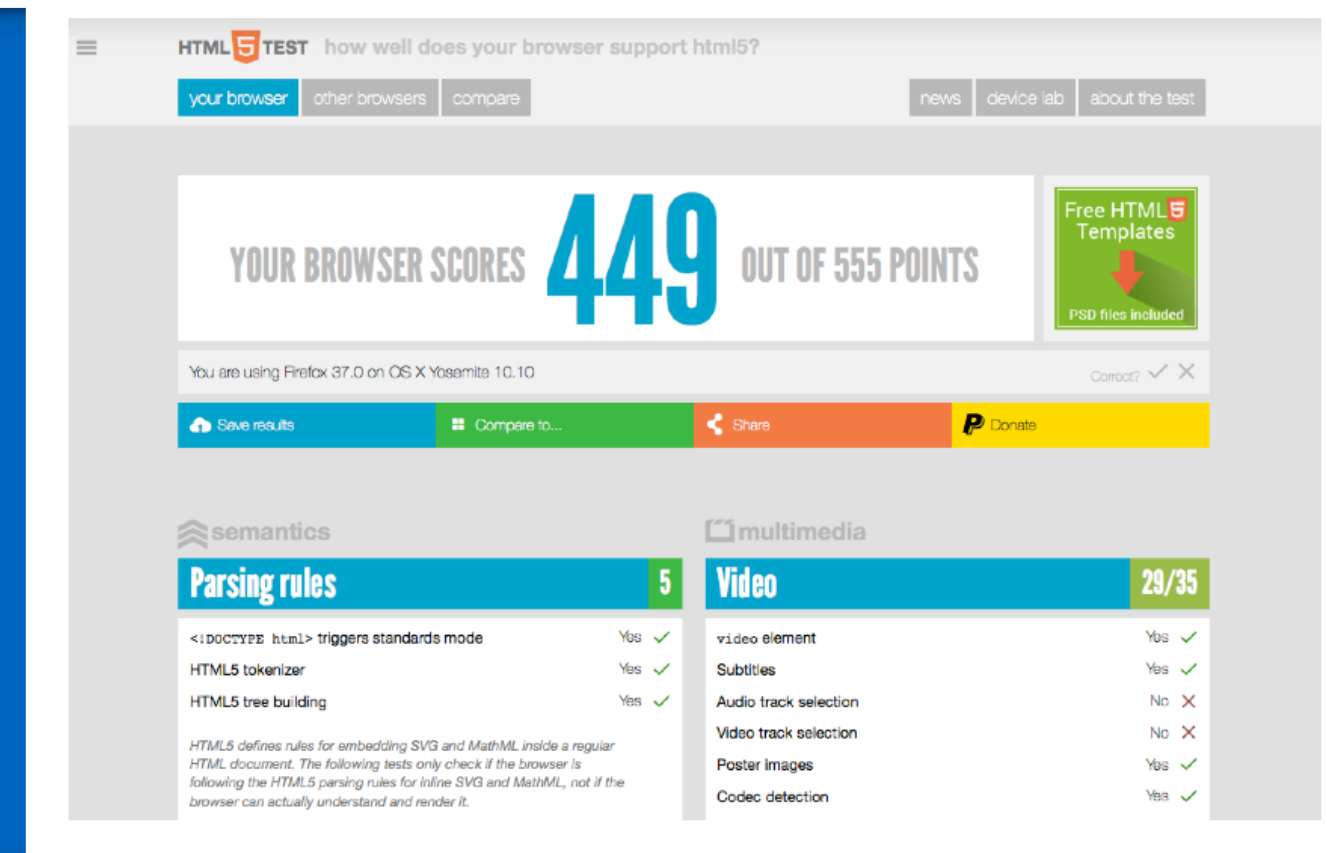
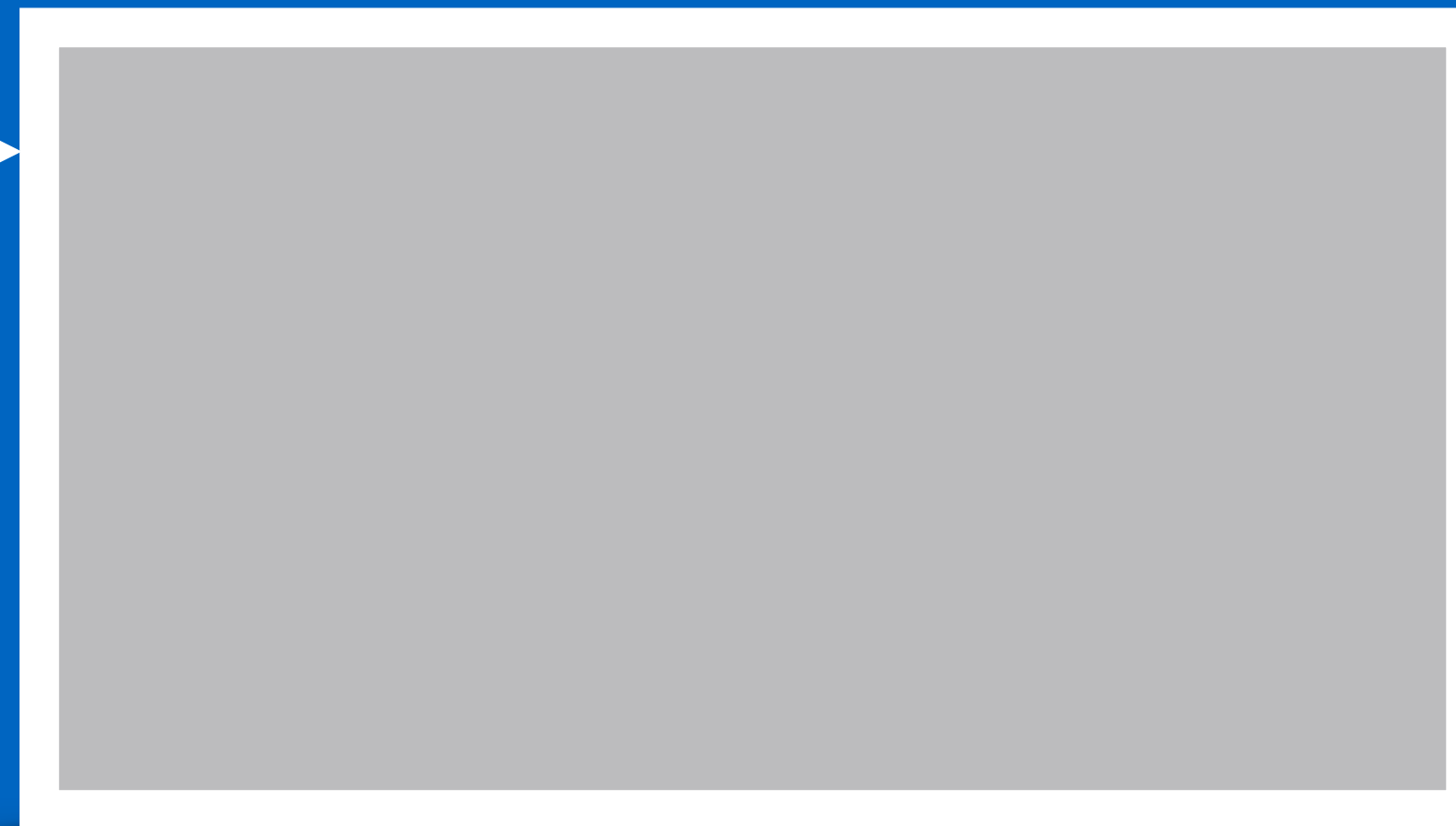


two screens  
(surprisingly normal)

# a dual visual viewport

(the bottom one is the primary visual viewport)

3d screen, but only  
2d is supported in  
the browser



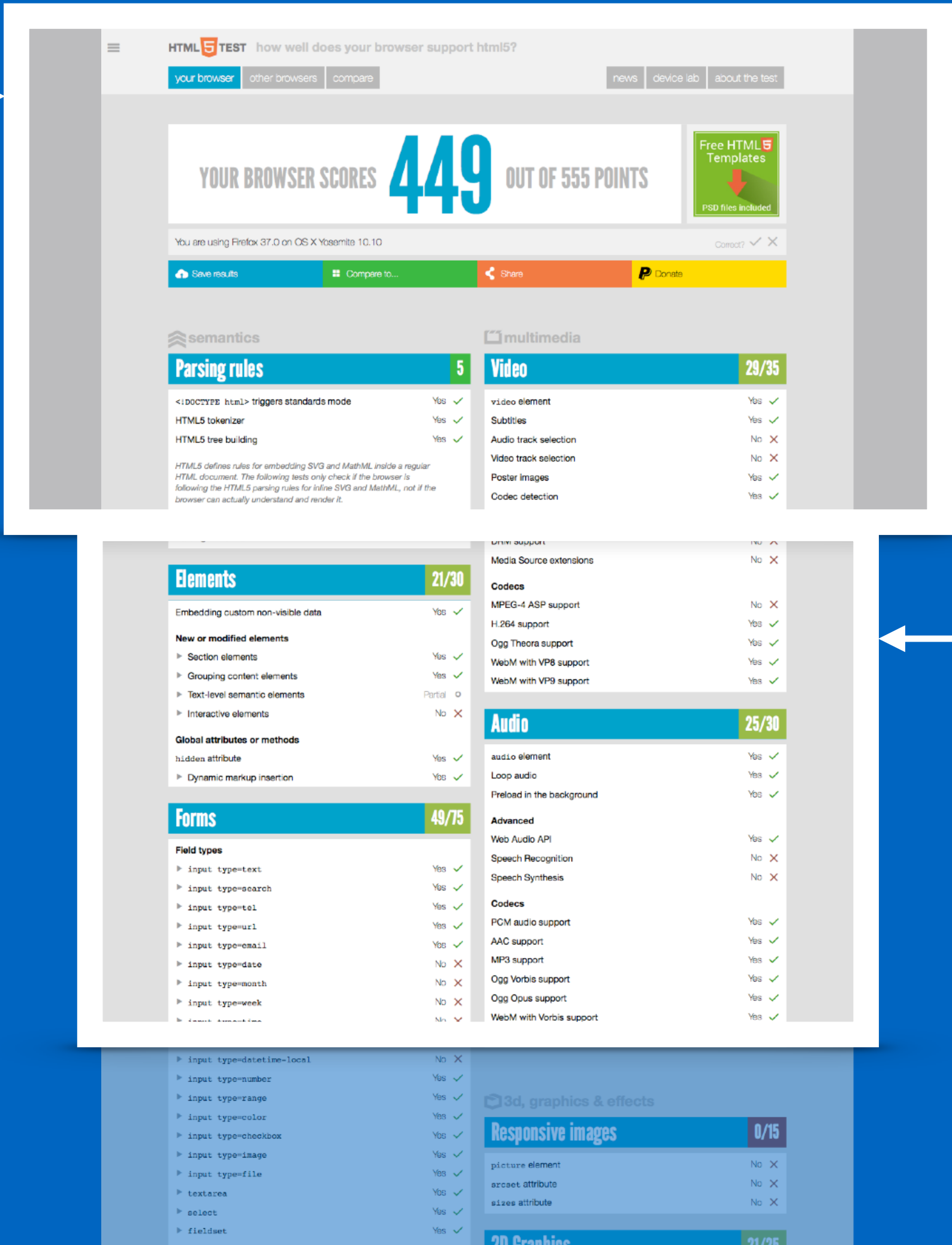
resistive  
touch screen



# a dual visual viewport

(the bottom one is the primary visual viewport)

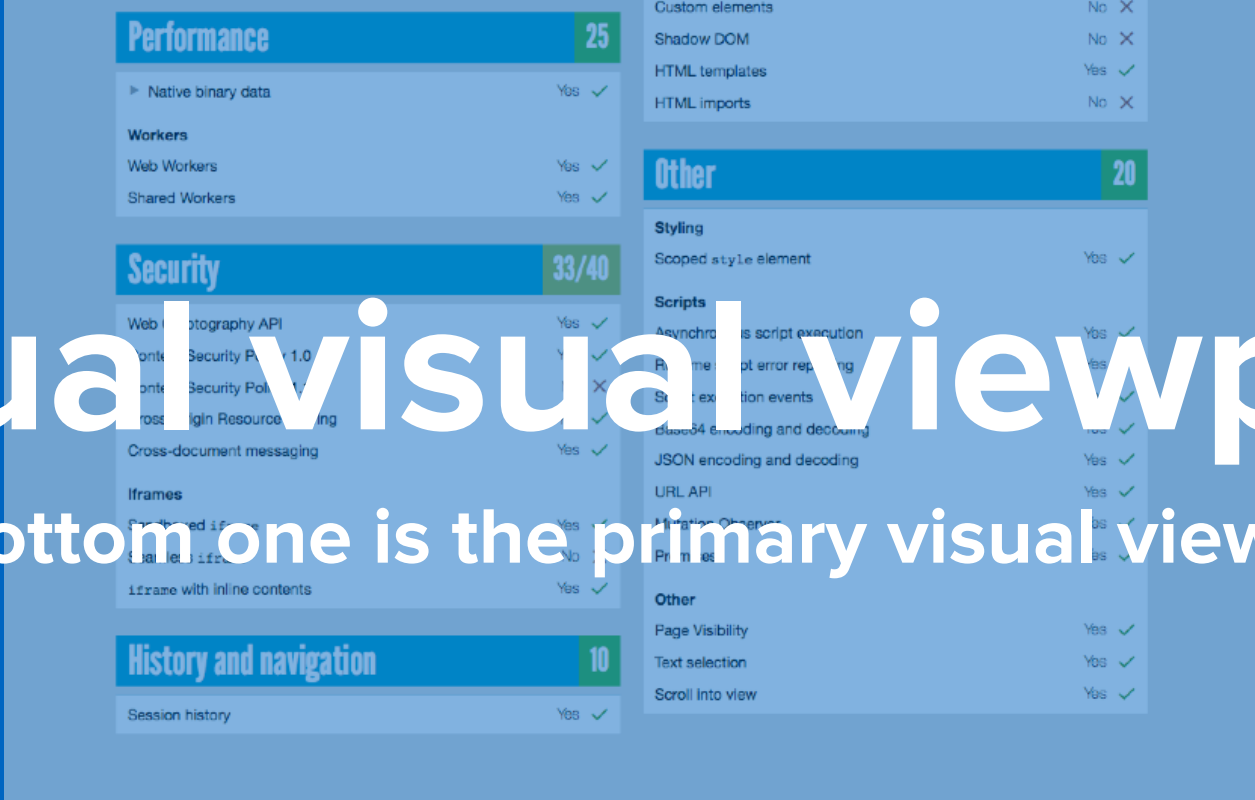
3d screen, but only  
2d is supported in  
the browser



resistive  
touch screen

# a dual visual viewport

(the bottom one is the primary visual viewport)



3d screen, but only 2d is supported in the browser



resistive touch screen

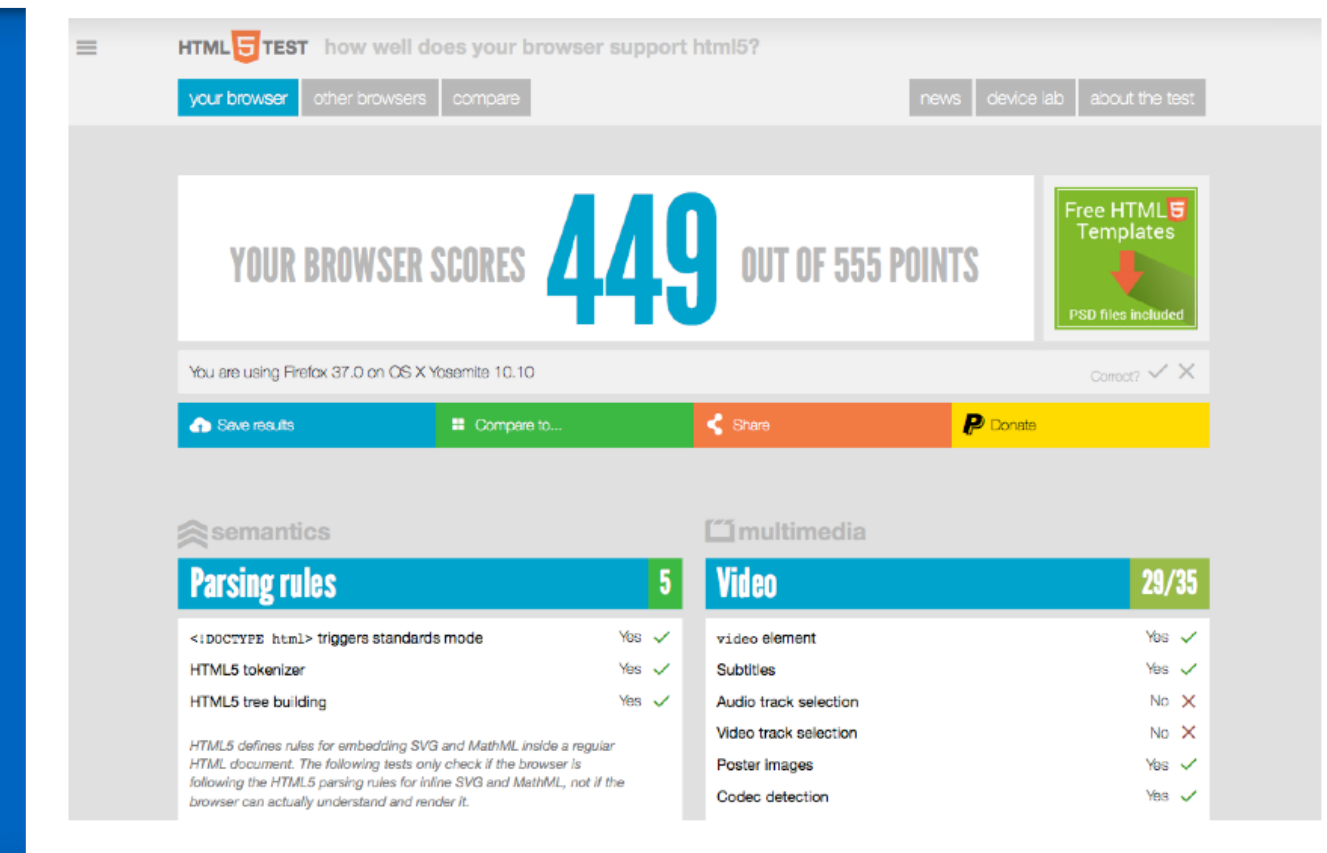
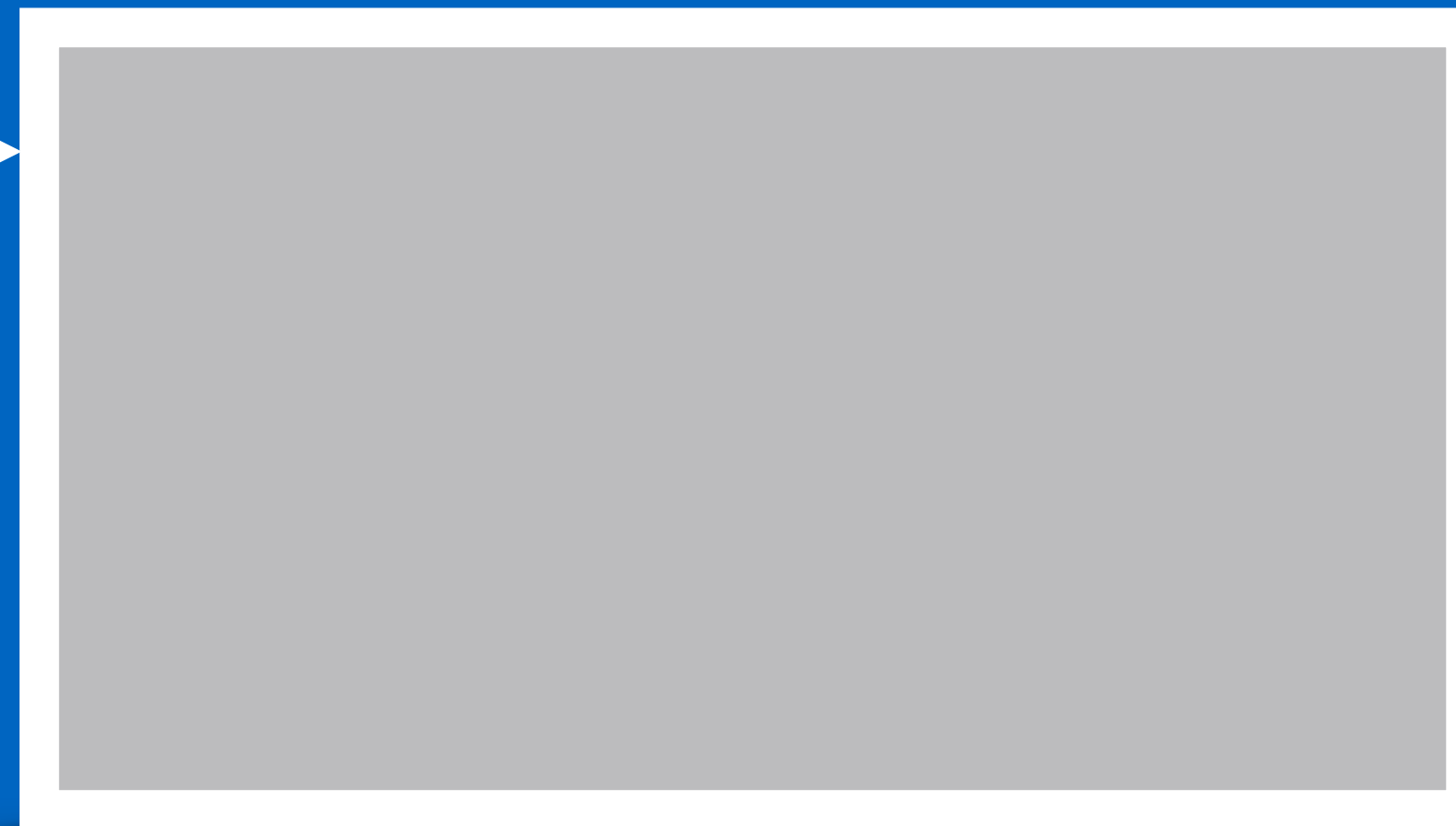




# a dual visual viewport

(the bottom one is the primary visual viewport)

3d screen, but only  
2d is supported in  
the browser



resistive  
touch screen









weird browsers!





“We cannot predict future behavior  
from a current experience that sucks”

– **Jason Grigsby**



but wait...



weird browsers!

~~weird~~ browsers!



rowsers

browsers

browsers

browsers

browser

browsers

browsers

browsers

users

browsers

browsers

browsers

brows



one arm



arm injury



new parent

permanent



situational





thank you

**niels leenheer**

@html5test