EVOLUTION OF WEBSITE LAYOUT TECHNIQUES

Maya Stoeva[1]

[1] Faculty of Mathematics and Informatics, University of Plovdiv “Paisii Hilendarski”, 24 Tzar Asen Str., 4000 Plovdiv, Bulgaria
Email: mstoeva@uni-plovdiv.bg

Abstract. We are now living, working, teaching, and learning in the constantly changing circumstances. Since the time HTML language was invented the layout technologies and techniques for website building changed a couple of times. To be a good web and UI/UX designer [1], front-end developer or teacher, who spreads the knowledge to the future specialists, you need to know which were and which are the current web technologies. Only in that way, you will be able to predict the future. The goal of this paper is to reveal the evolution of web layout techniques over the years and outline some of their future perspectives.

Key Words: *web layout techniques, web layout technologies, front-end frameworks, HTML tables, CSS grid, Float layout, Flexbox layout, responsive layout*

Introduction

Nowadays, even simple websites are kind of web applications. They can use almost unlimited possibilities for presenting a huge amount of information, structured semantically, including different kinds of interactive multimedia [1], [3]. More than 25 years ago, the situation was different. Tim Bernes-Lee invented the World Wide Web in 1989. Two years later, he proposed the first websites [5, 7, 8]. They were completely text-based. Through the years, designers and web developers have been trying a different kind of techniques to change the way websites work, look and feel. Meanwhile, technologies for building web layouts changed significantly, especially in the last ten years. Firstly, images were added to the page with headings and paragraphs. After that came table-based layout, Flash (currently known as Animate) interfaces, and CSS-based designs with three different techniques, which changed the whole approach. From the beginning to the present, web layout takes prominent progress in both perspectives – design and development. To be good specialists like web designers,

front-end developers, or teachers in the computer science field, it is good to know what is the evolution of these web technologies and techniques.

The main objectives of this contribution are:

1. Introduction to the important milestones in the web technologies evolution.
2. Introduction to different web layout building techniques, from the design and front-end development perspective.
3. The future of web layout design and implementation.

Evolution of web layout techniques

The techniques for building websites are determined by the available connected technologies. In the last 10-15 years, the latter has undergone many significant changes. For this reason, before these techniques are considered in detail, it is good to make a brief evolution overview of the most vital technologies milestones of the Web between its beginning and present. The table 1 below presents the most important events, which define the guidelines of the web interface development over the years:

Table 1. Evolution timeline of essential web layout technologies [1]-[8]

Period	Technologies
1991-1992	HTML 1.0 and WorldWideWeb (the first browser) were invented from Berners-Lee – *web sites contain only text with hyperlinks.*
1993-1994	Mosaic browser – *the first browser, which was able to display images inline with text information.*
1994-1998	1994 – *First web banner was placed.* 1995 – *GUI (Graphic User Interface) and UX (User Experience) were presented, together with Web-safe colors and JavaScript language (with an initial name Mocha), and Internet Explorer 1.0 and Netscape Navigator 2.0.* 1996 – *CSS 1 (Cascading Style Sheets, level 1) was released with official recommendation from W3C; Macromedia Flash 1.0 came with FutureSplash Viewer plugin.* 1997 – *HTML 3.2 came with table layout, although the <table> tag came with HTML 2.0 in 1995, together with . It was presented as a new important term Search Engine Optimization (SEO). JavaScript technology was combined with HTML and CSS to enhance website flexibility and interactivity and the result was Dynamic HTML (DHTML).*
1998-2002	1998 – *W3C presented the final recommendation for the Extensible Markup Language (XML) 1.0 specification. CSS 2 was released.*

	Google searching engine was invented from Larry Page and Sergey Brin, Ph.D. students in Stanford University. 1999 – *Web 2.0 was introduced like a term by Darcy DiNucci in the article entitled "Fragmented Future". The term describes development stage of the Web.* 2000 –*The Extensible HyperText Markup Language (XHTML) 1.0 specification was official published by W3C for creating web pages, which combine XML and HTML into three versions: strict, transitional, and frameset.* 2000 – *The first Media Queries specification was presented; Microsoft released Internet Explorer (IE) 6.0 as an integrated software into Windows XP operating system. IE 6 took more than 80% from browser's usage until 2004.* 2001 – *It was announced the graphical vector file format Scalable Vector Graphics (SVG) 1.0.*
2002-2006	2004 – *The technique called CSS Sprites was published by Dave Shea in the online magazine "A List Apart".* 2005 – *The YouTube website was launched and one of its co-founders, Jawed Karim, uploaded the first video called "Me at the Zoo" on April 23, 2005. This was the beginning of a social media revolution; Google presented its Google Analytics tool.* 2006 – *John Resig invented the jQuery JavaScript library, which is focused on the interactivity with users via JavaScript and HTML technologies; Hampton Catlin, together with Natalie Weizenbaum presented CSS preprocessor called Sass 0.1.0 (Syntactically awesome style sheets).*
2006-2014	2007 – *The first CSS Grid specification was released by W3C; BEM (Block, Element, Modifier), the CSS naming convention was developed.* 2009 – *It was released the next most famous CSS preprocessor: Less 1.0; W3C published the specification of CSS Flexible box layout (Flexbox), which was the third milestone for the web layout development after HTML tables and float techniques.* 2010 – *Google presented a paramount Google Web Fonts or shortly called Google Fonts; The term "Responsive Web Design" was announced by web designer Ethan Marcotte; Flat design 1.0 was promoted too; Mark Otto and Jacob Thorton, web developers of Twitter introduced Bootstrap 1.0 CSS framework.* 2012 – *The official recommendation for Media Queries specification was issued by W3C.* 2013 – *Facebook software engineer Jordan Walke developed first version of React JavaScript library or React.js/ReactJS; Bootstrap updated to version 3.*

	2014 – *This year was another milestone in the web layout development. Google presented a new design method called Material Design and W3C published the final recommendation for the HTML5 language.*
2014-2021	2015 –*The web browser Microsoft Edge was released from Microsoft, and Bootstrap was updated to version 4.* 2016 – *Flat 2.0 become a new important tendency.* 2017 – *CSS Grid Layout techniques became a new popular way for web site development.* 2020 – *Adobe Systems stopped the support of Adobe Flash Player.* 2021 – *Bootstrap updated to stable version 5*.

According to the author, the web interface development techniques are possible to be divided in three main stages: *before (until 2014)*, when final specification of HTML 5 was released, *current stage (until 2021)* and *future opportunities*. This chapter covers the first two.

Before

At the beginning of the Web, designers were too limited from the technologies. They were able to insert and to structure only text-based information, and the result was not pretty. Two years later, it was possible to integrate images, together with the text. However, web developers were still searching a way to structure the information in a better way. That became possible after **1997**, when **HTML tables** were released. In fact, they become **the first web layout building technique**. Tables were easy to implement and all browsers began to support them. They are used until now, mostly to develop newsletters. Table usage allows designers to create prettier pages, but are too heavy for loading. Years were passing and web community was searching for more elegant decisions. CSS language also were developing and in **2004** it was presented "float" CSS property [6]. It allows developers to make an easier transition to web layouts without tables. The float property became so preferred, because different block-level elements will not line up beside one another in a column-based format, at least by default. It allows developers to create table-like column layout in an HTML page without tables usage. **CSS float layout** is the second essential web layout technique. For more than 10 years it was the most used method for building web pages. The purpose of the CSS float property is to position a block-level element to the left or right side of the flow in relation to other block elements. This allows naturally-flowing content to wrap around the defined floated element. This idea is similar to the print materials, where images are aligned to the one side while the other content like text or another graphics flow naturally around the left- or right-aligned elements. Each floated element should have an explicitly set width, which ensures that the float behaves as expected and helps to avoid visual problems in certain browsers. Most of the layout issues with floats are usually fixed with usage of the CSS clear property. It lets developers to "clear" floated

elements from the desired side: from one or both sides of an element. In the web literature is possible to be found a lot of articles that described float-related issues in-depth [6]. At the same time, smartphones were becoming more and more usable and vary in size and technologies. However, that also caused problems. Developers and web communities were researching for new ways how to avoid many browsers' inconsistencies caused by float layout and display sizes. As a result, HTML 5 was released together with CSS 3 specification and the term for responsive web design was presented. These events were a prerequisite for the emergence of the other techniques for web layout building, used so far.

Current stage

Currently there are four main techniques to create web layouts:

- **CSS float property** (described above).
- **CSS flexbox** – CSS Working Group announced the idea for the flexible box model a long time ago - before 2008. Their original specification is based on XUL, the technology Firefox uses to produce its UI. It contained layout primitives that were easier to use than the plain HTML when building web interfaces. In the beginning, the layout algorithm was slow and worked differently between Webkit and Firefox, so it was not the best choice for the developers. In 2011, Tab Atkins did a significant change of the syntax, which was supported later in Chrome, Opera, and IE 10. In the next 4 years, Flexbox specification was changed a couple of time. As a result, from May **2015** until now Flexbox layout technique is one of the most used. It is also known as one-dimensional layout, because is based on "flex-flow directions" and ordered elements in rows or in columns [1, 2, 8]. Flexbox is not only a single property, but a whole module. It contains a lot of objects including their own set of properties. Some of the elements are set to be the container – a parent element, called "flex container", whereas the others are defined as children or "flex items". Flexbox layout is faster than the float one, because it provides one more efficient way to layout, align and distribute space among items in one container independent of their size, or dynamic. The main aim of the flexbox layout is to give the containers ability to alter their elements' width/height to fill the available space in the best possible way. A flexbox container expands items and fill all available free space or shrinks them to avoid overflow. This functionality allows developers to build a responsive layout with minimum amount of writing additional CSS media queries. Flexbox layout is most used to building interfaces of web application or small web pages. That is the reason, the author to use this technique in the exercises of "Web design" university course [5] for websites implementation, in last two years.

- **CSS grid** – The **CSS grid layout technique** is much newer (from **2017**) in the practice than Flexbox and still has not enough browser support, although like an initial specification it is older than Flexbox (from 2007). Furthermore, grid can do things Flex, that is not able to do and vice versa, but surprising a grid object can be a flexbox container, and a flexbox – a grid container. CSS grid is also known as a two-dimensional layout, because distributes elements both in rows and in columns [5, 6, 8], and often is used for larger scale interfaces. This is the newest way for building layout. The CSS Grid layout offers a grid-based layout system, build with rows and columns. It provides a method for web pages design without need using floats for positioning. It owns positives and negatives sides, but the initial idea is similar to the table layout. So, a big part of web developers think that its usage is equal to turn back in the 1997 year. CSS grid layout specification is still in progress but is more robust now, and we can use it without need to make additional code adjustments to accomplish one basic layout.
- **CSS framework** – When developers want to speed up web layout development, they can use CSS frameworks. The most famous of them are Bootstrap, W3.CSS, Pure CSS, Foundation, Skeleton, Materialize, Semantic UI, UIKit, and Miligram [6]. All of them are modern, responsive, mobile-first CSS frameworks. They provide to their users a ready basic layout with embedded responsiveness. No matter which one the developer will choose, he/she needs to check which layout technique comes with the framework. For example, Bootstrap 3 uses a float layout, but Bootstrap 4 – a flexbox. It is strongly recommended to avoid mixing two types of layout techniques because that will lead to unpredictable visual results in browsers.

The future web layout design and development

The Web itself is relatively short and too many technologies have been born and have died in a short period of time [6, 7]. Aforetime, designers and developers were obstructed by technology, but now we are at a stage when only their imagination is their limitation. So, the future of web layout is no longer about what can be done, but rather about what should be made. It is time to turn more attention to user experience acceptance of the web content and functionality. Web designers should consider carefully how their design can and will affect the users in positive experiences. Web developers need to continue creating fast-loading and simple web applications with a well-defined structure [9-11]. The virus Covid-19 changed the working and learning environment. Most people are operating online, so more and more businesses depend on the performance of their websites. That provoked one new design type – Lead-Driven web design [6]. That is a data-guided approach. It combines analytical data, user intent, and behavioral analysis to create websites and applications that generate qualified leads that convert into engaged customers. The evolution of social media and connected data

and measurement tools have become more and more popular. So, this Lead-Driven web design is still gaining popularity.

Other future technology changes will be focused on:

- Creating of more conversational user interfaces that fulfill a conversation with an actual human or chatbot support usage.
- Additional usage of Augmented Reality (AR) and Virtual Reality (VR).
- Developing much more accessible websites to people with different kind of disabilities. Voice interface implementation can help a lot for that.
- In all over the world cyber-crime attacks are increasing ten times more in the recent years. Artificial Intelligence (AI) and Machine Learning (ML) can help to prevent some of them.

The world is changing too fast, web technologies too. It is becoming harder and harder to follow all new things. That is why is so important to know what was before and what is the current state of web techniques. If we do that constantly will be easier for us to develop web layouts, which are completely user-oriented and correspond to users' needs.

Acknowledgments

I want to thank all the members of the "Computer technology" department for the teamwork and the help they have given me over the years. Special thanks to Prof. Asen Rahnev, Prof. Angel Golev, Prof. Anton Iliev, Assoc. Prof. Nikolay Pavlov, Assoc. Prof. Kremena Stefanova and Assoc. Prof. Evgenia Angelova.

References

[1] B. Frain, *Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques, 3rd Edition*, Packt Publishing Ltd., Birmingham B3 2PB, UK, 2020, ISBN: 978-1-83921-156-0.

[2] J. Yablonski, *Laws of UX*, O'Reilly Media, Inc., USA, 2020, ISBN: 978-1-492-05531-0.

[3] B. LaGrone, *HTML5 and CSS3 Responsive Web Design Cookbook*, Packt Publishing Ltd., UK, 2013, ISBN: 978-1-84969-544-2.

[4] W. Galitz, *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, Third edition*, Wiley Publishing Inc., USA, 2007, ISBN: 978-0-470-470-05342-3.

[5] T. Glushkova, M. Stoeva, *Lectures and exercises from "Web design" course in FMI, University of Plovdiv "Paisii Hilendarski"*, 2020/2021,

https://innotechtoday.com/the-future-of-technology-in-web-design/, last accessed 2021/18.

[6] Web layout design types, *https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Introduction, https://www.w3schools.com/html/html_layout.asp, https://www.smashingmagazine.com/category/css/, https://xd.adobe.com/ideas/principles/web-design/11-website-layouts-that-made-content-shine-in-2019/, https://medium.com/@ux_hai/flat-2-0-why-fully-flat-design-is-outdated-95fb5609f729, https://alistapart.com/article/responsive-web-design/, https://alistapart.com/article/the-story-of-css-grid-from-its-creators/, http://info.cern.ch/hypertext/WWW/TheProject.html, https://www.smashingmagazine.com/2009/10/the-mystery-of-css-float-property/, https://annairish.github.io/historicizing/history, https://www.lambdatest.com/blog/best-css-frameworks-2021/* last accessed 2021/18.

[7] Dreamscape Design, *The Evolution of Web Design*, Digital Agency Network 2021, *https://dreamscapedesign.co.uk/web-design-evolution*, last accessed 2021/18.

[8] Web design layout history, *https://www.webdesignmuseum.org/web-design-history, https://quibble.digital/the-evolution-of-web-design, http://grid-layout.com/history.html, https://css-tricks.com/snippets/css/a-guide-to-flexbox/*, last accessed 2021/18.

[9] Evan, R., *The Future of Technology in Web Design*, Innovation & Tech Today, 2021, *https://innotechtoday.com/the-future-of-technology-in-web-design/*, last accessed 2021/18.

[10] Troncoso, D., *The Future of Web Design*, MarketSplash 2021, *https://marketsplash.com/future-of-web-design*, last accessed 2021/18.

[11] King, E., *What is the Future of Website Design: How Will Web Design Change?* Digital Agency Network 2021, *https://digitalagencynetwork.com/what-is-the-future-of-website-design-how-will-web-design-change*, last accessed 2021/18.