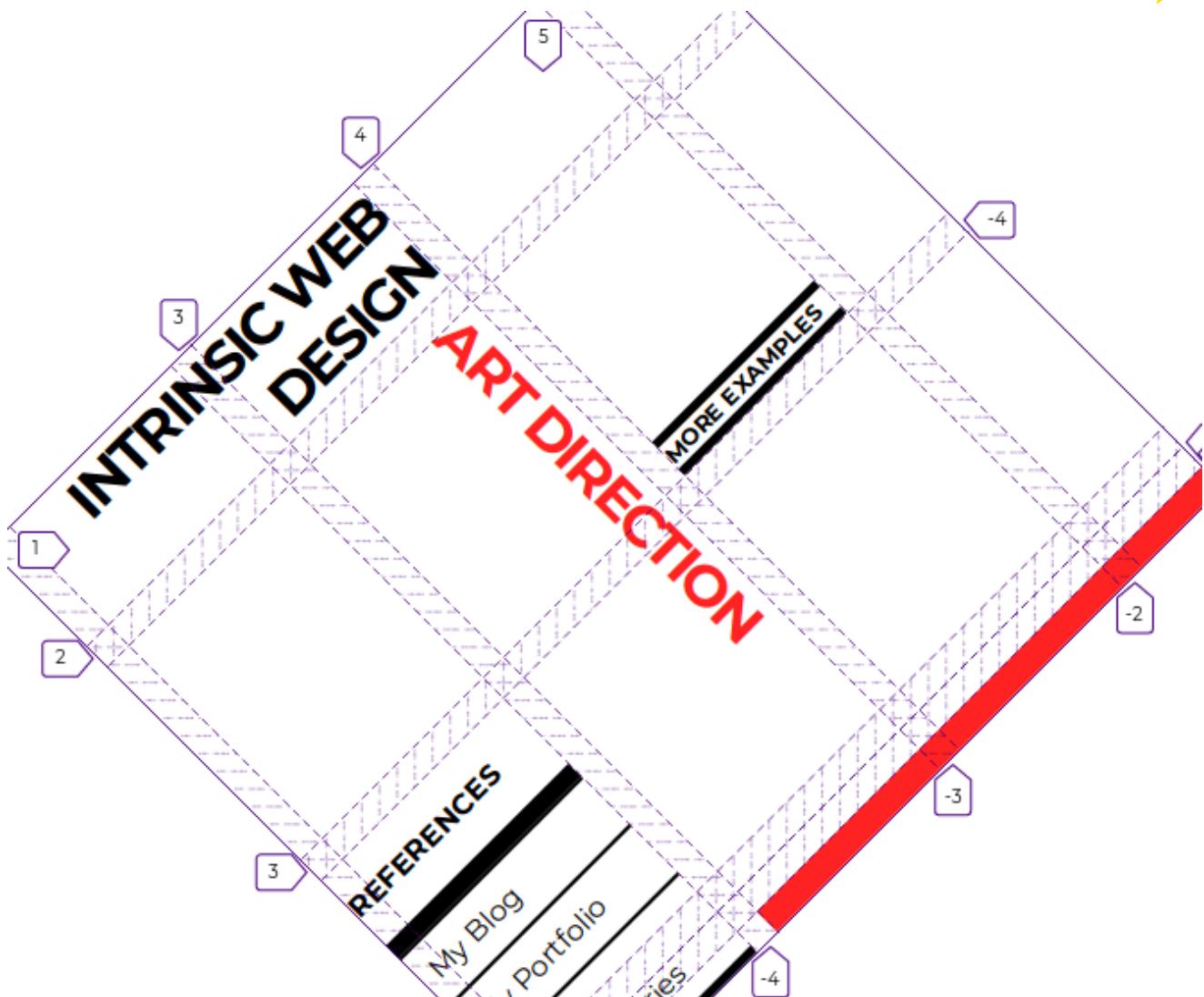


# antares:



Featured

## Intrinsic Web Design: Ito na ba ang Wakas ng Responsive Web?

<https://celestialcinnamon.github.io/antares-blog/tl/Intrinsic-Web-Design-Ito-na-ba-ang-Wakas-ng-Responsive-Web/>

Francis Rubio 12-23-2018

Mainit na mainit sa mga Web designers at developers ang RWD o Responsive Web Design. Pero simula noong 2018, may bagong buzzword na binabato, isang technique na papalit sa Responsive Web Design: ang *Intrinsic Web Design*. Dapat mo ba itong pag-aralan, o isa lang ito sa mga bagong uso na malalaos din pagkatapos ng dalawang taon? Sulit bang pag-aralan ang *Intrinsic Web Design*?

Isang taon na mula nang unang marinig ng mundo ang terminong *Intrinsic Web Design* galing kay Jen Simmons sa An Event Apart, pero sa hindi pa ito gaanong matunog sa Pilipinas (or at least sa mga kakilala ko). Ito, ayon kay Jen Simmons, ang papalit sa nakasanayan nating Responsive Web Design. Sabi pa niya, isa itong bagong era ng Web design. Ano ang ibig niyang sabihin?

## Ang Iba't Ibang Era ng Web Design

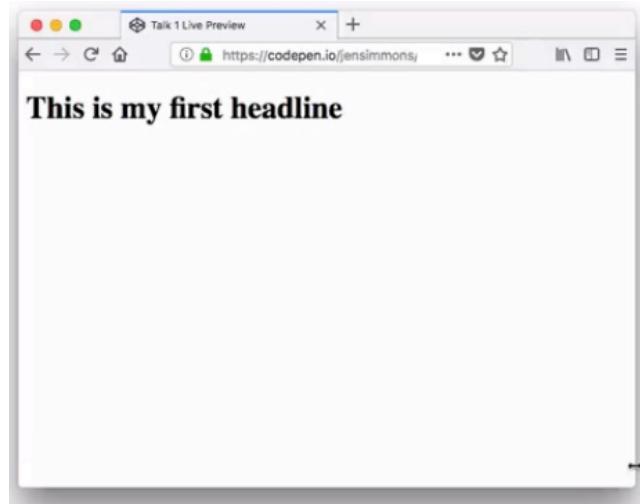
Matagal nang umiiral ang World Wide Web, at malayo na ang narating nito simula nang maimbento ito ni Tim Berners-Lee noong 1990.<sup>1</sup> Naging platform

din ito para sa mga artist na nagnanais maglabas ng mga gawa nila sa mundo, o sa mga kompanyang nagnanais gawing artistic ang Web sites nila para umayon sa branding ng kompanya.

Pero hindi hawak ng designers at developers ang lahat ng tools na magagamit para makagawa ng perpetong design. Kaya naman pumasok tayo sa iba't ibang era ng Web design. Bawat isang era ay may natatanging technique o hack na ginagamit para magawa ang design.

### Simpleng HTML

**Defining Characteristic:** Puro text ang mga Web page at walang images.



Kuha mula sa *slides ni Jen Simmons*

Nang magsimula ang Web, isa lang ang language nito: Hypertext Markup Language. Masasabing hindi ito para sa mga designers. Sa isip ng creator ng Web na si Tim Berners-Lee, hindi mga author, kundi mga user ang may kapangyarihan sa kung ano ang magiging hitsura ng mga Web page. Kaya naman hindi noon kinailangan ng anumang design tools sa Web. Lahat ay text. At dahil isang *markup language* ang HTML, ipinapakita lang nito ang structure ng isang Web page, meaning ng bawat bahagi ng isang Web page. Para kay Tim Berners-Lee, dapat kontrolado ng users kung ano ang magiging hitsura ng Web pages sa pamamagitan ng customizations.

## Flow Layout

**Defining Characteristic:** Gaya sa Simple HTML era, pero ngayon ay may images na, at nababago na rin ang mga background color at text color. Mas makulay na ang buhay.



Nagsimula ang design sa Web nang magsimulang magkaroon ng images sa Web (salamat sa Netscape Navigator). Sa panahong ito kung kailan lahat ay text at HTML lang ang mayroon tayo, iisa lang ang uri ng layout—ang tinatawag nating flow layout. Dumadaloy ang text mula kaliwa pakanan, at kapag wala nang space, ang susunod na character ay magsisimula sa susunod na line.

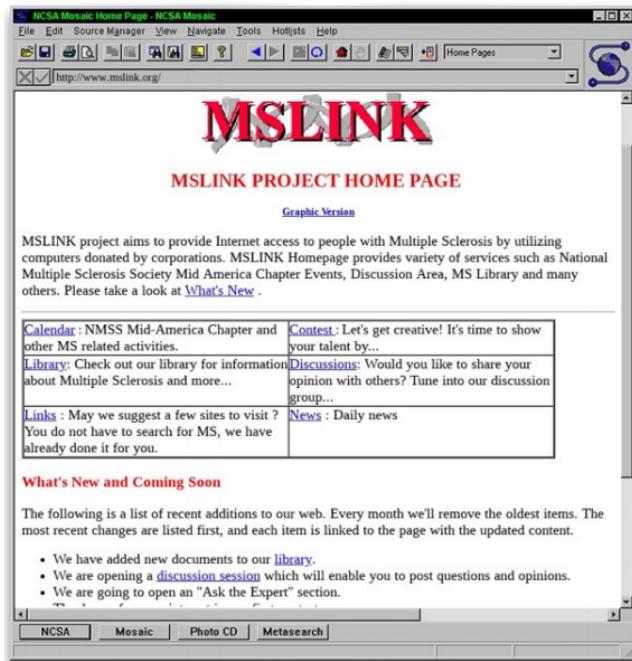
Ito rin ang panahon kung kailan `<hr>` lang ang pinaka-design na mayroon sa Web. Kung design lang din ang pag-uusapan, walang gano'n sa Web sa panahon ng flow layout era. At tama lang dahil sa progreso

ng teknolohiya sa panahong iyon,  
napakabagal ng Internet, kaya naman mas  
praktikal kung text lang talaga ang Web.  
At bukod pa ryan, karamihan ng users ng  
Web sa panahong ito ng Web design ay  
puro mga teknikal at propesyunal.



**Figure 20.5827**

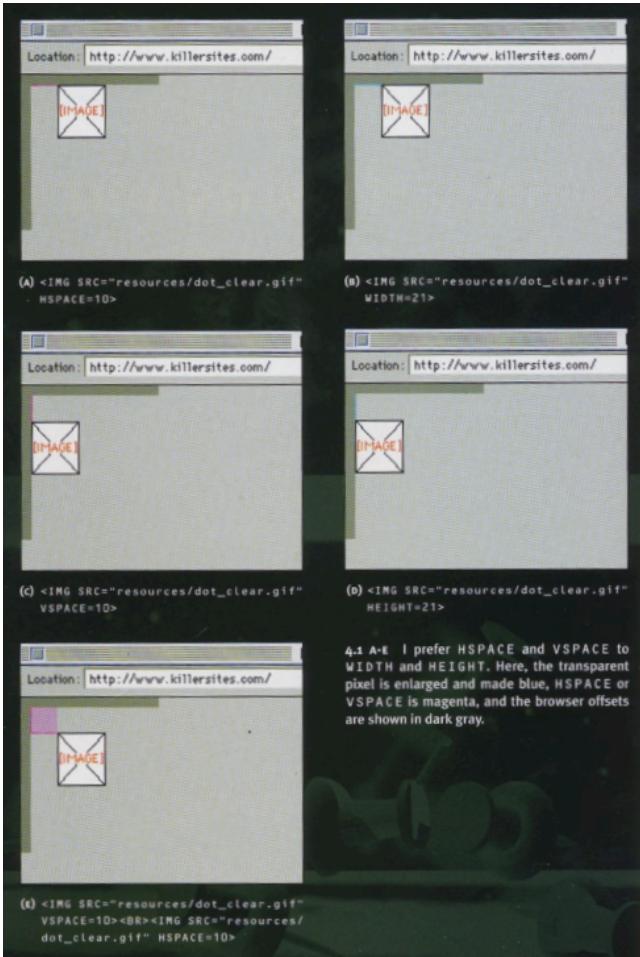
Gaya ng makikita sa [figure 20.5827](#), nag-exist noon ang mga center-aligned na pages (gamit ang lumang HTML tags na `<center></center>` na sana wala nang gumagamit ngayon).



**Figure 60.7538:** Gumamit din ng tables para sa layout noong Flow Layout Era, pero hindi kasingdalas gaya ng sa Table Layout Era.

## Table Layout

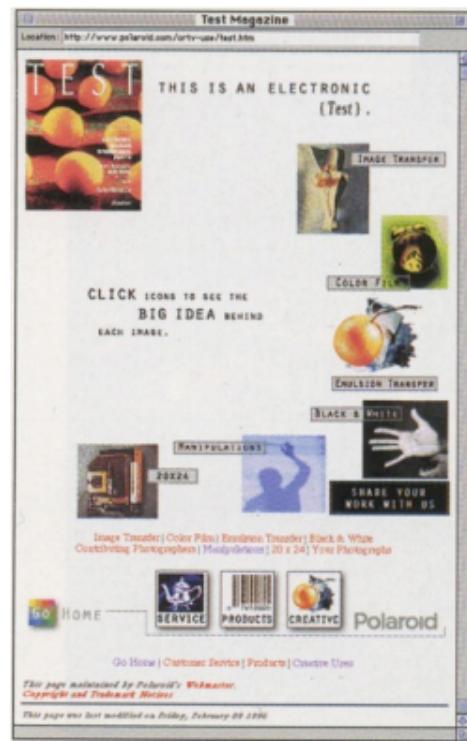
**Defining Characteristic:** Malawakang paggamit ng `<table></table>` tags. Bagaman ginagamit ito sa panahon ng flow layout, mas malawakang ginagamit ngayon ang HTML tables, at kadalasan, para lang sa layout sa halip na mga data na nasa table.

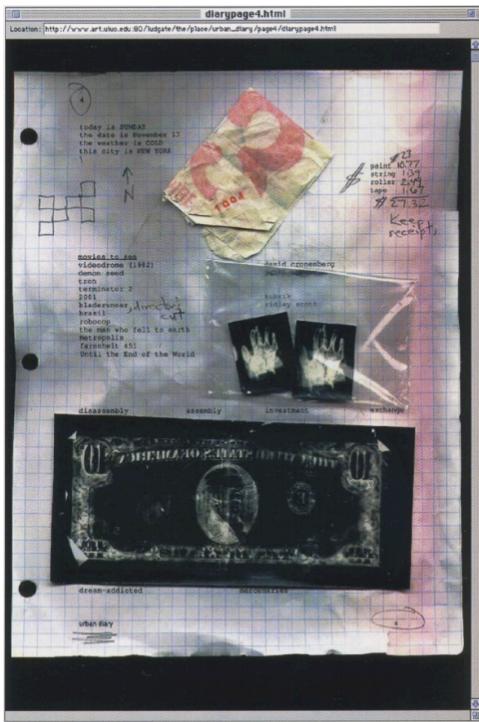


**Figure 86.6879:** Mula sa 1996 na aklat ni David Siegel, *Creating Killer Websites*

Sa era na ito ng Web design, pumasok na tayo sa isang panahon kung kailan untilti nang dumarami ang users ng Web. Masasabing dito rin nagsimula ang “design” sa Web, as in design na gaya ng alam natin sa ngayon. Gumamit ang mga Web developers ng <table> tags para gumawa ng mga layout sa mga Web page nila. Gumamit din sila ng mga tinatawag

na spacer GIFs, i.e., mga GIF images na 1px×1px ang laki para lang magkaroon ng laman ang mga table cells at maging maayos ang mga layout ng Web pages.





**Figure 73.1475:** Mapapansing nagsisimula nang maging mas moderno ang artistic approach sa Web. Malayong-malayo sa orihinal na daang tatahakin sana ng Web design.

Dito rin nauso ang proseso tinatawag nating *slice and dice*. Ang mga design ay ginagawa sa photo-editing software gaya ng Photoshop. Pagkatapos, ang mga design, o kung minsan ay mga background image, ay kina-crop at pinaghihiwahiwala. Bawat isang piraso o slice ay mapupunta sa isang table cell hanggang sa mabuo ang Web page.

## Flash

## Defining Characteristics: Paggamit ng Adobe Flash para sa mga Web design.

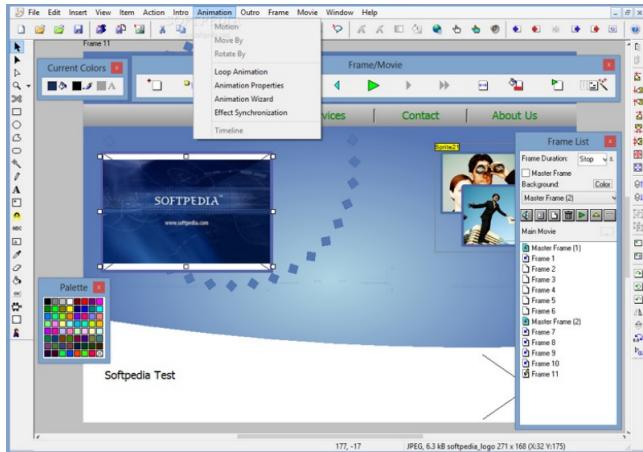


Figure 93.6485

Sa panahong ito, lumayo ang Web mula sa pinanggalingan nitong HTML at nakakita ng bagong tahanan sa Adobe Flash.

Kumpara sa HTML tables, mas flexible nang di-hamak ang Flash, lalo na kung isasama rito ang animations. Para sa mga designers, ito na ang sagot sa ilang dekadang paghihirap na makagawa ng mga design. Pero gaya ng alam mo siguro, sa paglipas ng panahon nilayuan ito ng mga tao dahil bukod sa mabagal ito, kinakailangan ng mga users na mag-install ng plugin sa browsers nila para lang makita ang mga Web page. Hindi rin ito accessible, kaya nahirapan ang mga may

kapansanan na gamitin ang marami sa mga Web site. At bukod pa ryan, napakalaking problema nito sa security ng users. Kaya naman papatayin na ng Adobe ang Flash sa 2020.

## Ang Simula ng CSS Epoch

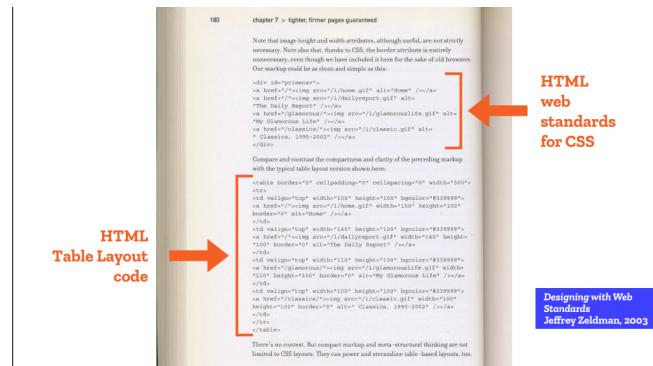


Figure 93.6485: Kapansin-pansing dahil sa CSS, nanumbalik ang tamang kahulugan ng mga HTML tags. (Mula sa 2003 aklat ni Jeffrey Zeldman, *Designing with Web Standards*)

Maaring imbento ko lang ang terminong CSS Epoch, pero marami ang sasang-ayon na nagsimula ang isang bagong panahon para sa Web platform nang dumating ang Cascading Style Sheets o CSS. Ito ang nagsalba sa Web design. Dahil sa CSS, mas maraming posibilidad ang nabuksan, kasama na ang accessibility at semantics ng Web sites. Ang mga sumunod na era

pagkatapos ng simula CSS epoch ay mas mabuti kaysa sa mga naunang era dahil lang sa mga posibilidad na nabuksan.

## Fixed-Width vs. Fluid Layout

**Defining Characteristic:** Dalawang magkalabang opinyon: fixed na width at percent-based layouts.

Dito sa era na ito, nagkaroon ng dalawang magkalabang opinyon sa Web design. May mga nagsasabi na dapat nating gawing fixed-width ang mga layout, gaya sa mga printed materials. Dahil kahit na kakaunti lang ang iba't ibang screen sizes noon, hindi pa rin natin kayang tiyakin kung ano'ng size ng screen ng mga users. Kaya dapat nating gawing naka-fixed ang laki ng layout.

Pasensya na, hindi mapapanood ang videos sa papel!

**Figure 62.5674 (video):** Isang halimbawa ng fluid layout.

May mga nagsasabi naman na "hindi, dapat nag-adapt sa size ng screen ang layout." Dapat na nagre-resize din ito

depende sa viewport. Hindi ako eksperto sa kasaysayan ng Web design, pero sa tingin ko, ang fluid layout ang naging precursor ng Responsive Web Design.

Pasensya na, hindi mapapanood ang videos sa papel!

**Figure 62.5674 (video):** Isang halimbawa ng fixed layout.

Dalawang naglabang opinyon, pero parehas na may punto. Sa era na ito, nagkaroon ng kritikal na desisyon kung aling direksyon sa sining ang tatahakin ng bawat designer, at kung paano ito ie-execute ng mga developer.

## Responsive Web Design

**Defining Characteristic:** Iba't ibang layout depende sa screen size.

Malawakang paggamit ng mga image na nagre-resize kasama ng viewport, mga media queries, at float elements.



**Figure 93.6485:** Ang Responsive Web Design ay isang flexible grid (kasama ang mga flexible na image) na gumagamit ng mga media query para makagawa ng isang layout na nagbabago sa iba't ibang kalagayan. (*Mula sa slides ni Ethan Marcotte para sa kaniyang pahayag sa An Event Apart 2010*)

Ang era na ito ang pinakakilala ng marami sa atin sa ngayon. Ito ang era ng float-based layouts. Sandamakmak na `float: left` at mga `clearfix` ang ginamit natin sa mga design para makagawa ng mga layout. Ito ang era ng 12-column layouts. Dito rin sumikat ang mga frameworks na gaya ng Bootstrap, Foundation, 960 grid, at iba pa.

Sa panahong ito rin halos nawala ang “art direction” sa Web. Dahil karamihan ng Web sites ay gumagamit ng iyon at iyon

ding frameworks na may 12-columns, naging magkakamukha silang lahat. Nalimitahan ang pagkamalikhain sa Web. Pero ang maganda naman dito, lumitaw ang mobile-first designs at gumagana sa lahat ng device ang mga Web site.

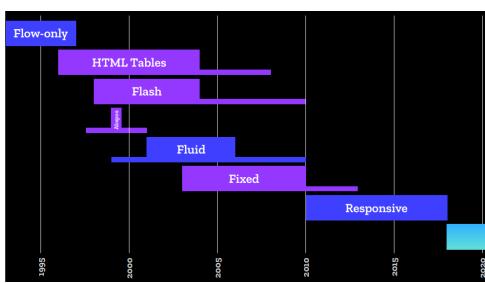
Hindi rin siguro ito matunog sa mga kakilala ko, pero mga ilang taon pagkatapos sumikat ang Responsive Web Design, lumabas ang Adaptive Web Design. Hindi nito pinalitan ang RWD; sa halip, dinagdagan nito ang RWD ng maraming kakayahan. Sa RWD, lahat ng transformations sa layout at adjustments ay nangyayari sa browser, gamit ang CSS, at kung minsan, JavaScript. Pero nang dumating sa eksena ang Adaptive Web Design, sumama ang server sa pagbibigay ng magadang experience sa mga user.

Pasensya na, hindi mapapanood ang videos sa papel!

**Figure 62.5674 (video):** Isang halimbawa ng isang responsive na layout.

Dahil mobile-first na ang Web, at mabagal ang Internet speeds para sa mga mobile

devices, hindi puwedeng mag-send sa mobile devices ng mga high-resolution images gaya ng ginagawa sa mga desktop. Kaya nagde-decide ang server kung aling size ng image ang ipapadala nito sa device depende sa width ng viewport nito. At hindi lang ito natatapos sa width ng mga images, dahil may mga sites na gumagawa ng dalawang magkaibang site para sa desktop at sa mobile para mabawasan ang file size ng CSS nila sa mobile devices.



Ito ang mga era ng Web design na alam natin sa ngayon. Posibleng may mga overlaps at mga hindi ako nabanggit, pero sa pangkalahatan, ang punto nito ay maraming pinagdaanan ang Web para makarating sa kung nasaan man ito sa ngayon.

## **Ang Pagpasok ng Intrinsic Web Design**

# **intrinsic**

(adj.) belonging to a thing by its very nature. Inherent. Innate

Ang pangako ng Intrinsic Web Design ay nasa pangalan na mismo nito: **intrinsic**. Hindi na natin kailangan ng frameworks para makagawa ng mga komplikadong layouts. Hindi na natin kailangang gumamit ng mga CSS properties na ginawa para sa ibang dahilan para lang magawa ang gusto nating design. **Ang Intrinsic Web Design ay isang improvement sa Responsive Web design; samakatuwid nga, Responsive Web Design +.**

Isang halimbawa nito ang `float` property sa CSS. Simula noong Table Layout era, nakabase na ang mga design ng Web pages sa mga grid. At kahit sa paintings at ibang artworks, malawakang ginagamit ang

mga grid. Pero dahil sa kakulangan ng features ng CSS, hindi natin ito replicate nang maayos. Kaya gumamit tayo noon ng `<table>` tags, at nang magtagal, ginamit natin ang `float` property para gumawa ng mga “column”. Matrabaho ito, oo, pero mas mabuti na ito kaysa sa paggamit ng `<table>` o ng `<frameset>`, na parehong nagpapahaba ng markup.

Pero dumating sa mga browser ang `display: flex` at `display: grid` features ng CSS, at sa wakas, nagkaroon ng tunay na grid system sa Web. Kaya naman hindi na natin kailangang gumamit ng `float` (na ginawa para sa ibang mga bagay) para gumawa ng mga columns. At ang maganda pa rito, dahil sa CSS Grid, hindi na tayo naka-focus sa columns; kaya na rin nating manipulahin ang rows. Ang CSS Grid ang pinaka-powerful na feature ng CSS sa ngayon.

## **Talaga Bang Papalitan na ng Intrinsic Web Design ang Responsive Web Design?**

Kung titingnan, nalalaos lang naman ang

isang technology kapag wala nang gumagamit nito. Sa ngayon, talamak pa rin ang RWD sa mga website design. Kaya parang mahirap isipin na malalaos ang RWD; kung sa bagay, napakaraming magagandang dulot ang RWD sa mga user. Pero tingnan natin ang mga pagkakaiba ng IWD at RWD—mga dahilan kung bakit naisip ni Jen Simmons (at kung bakit sinasang-ayunan ko) na Intrinsic Web Design ang future ng Web design.

### **Images**

**Responsive:** sa Responsive Web Design, lahat ng images ay flexible at nagrespond sa lapad ng viewport. Samakatuwid nga, lahat ng image ay percent-based.

**Intrinsic:** Sa Intrinsic Web Design, itinataguyod ni Jen Simmons na gumamit ang mga Web designer at developer ng flexible images kasama ng mga fixed-width images. Hindi na kailangang lahat ay flexible; puwede ka nang mag-set ng width, height, o iyong dalawang iyon sa mga image. At ang maganda pa rito, puwede ka na ring gumamit ng mga image

na flexible pero magiging fixed depende sa size (hindi lang width) ng viewport.

## Grids

**Responsive:** sa RWD, gumamit tayo ng pseudo-grid; pero sa totoo lang, hindi naman ito grid dahil columns lang ang ginamit natin. At sa totoo lang din, hindi rin ito mga columns kundi mga pseudo-columns dahil hindi naman iyon mga tunay na column. Ginawa sila gamit ang `float` property sa CSS, na hindi naman ginawa para sa mga columns. Sa madaling sabi, nagtagpi-tagpi lang tayo ng features para mapunan ang gusto nating gawin na wala sa CSS.

**Intrinsic:** sa IWD, lahat ay tunay: ang grid na gamit natin ay totoong grid dahil may rows at columns na ito. At hindi tulad sa RWD, ang mga rows at columns ay puwedeng fixed-width, o nakadepende sa size ng laman nito, sa halip na flexible lang at percent-based. Katunayan, hindi na lang percent ang puwedeng gamitin; puwede na rin tayong gumawa ng ratio-based columns at rows sa tulong ng `fr` units.

## Media Queries

**Responsive:** sa RWD, kailangan ang media queries sa lahat ng panahon para makagawa ng mga layouts na nagrespond sa width ng viewport.

**Intrinsic:** sa IWD, hindi kailangan ng media queries sa lahat ng pagkakataon. Dahil sa kapangyarihan ng CSS Grid, makakapag-adjust sa width ng viewport ang width ng layout.

## Flexibility

**Responsive:** sa RWD, flexible ang layout sa diwa na may iba't ibang hiwa-hiwalay na layout para sa iba't ibang screen.

**Intrinsic:** sa IWD, flexible ang layout sa diwa na hindi kailangan ng hiwalay na layout para sa ibang content. Sa Intrinsic Web Design, hawak mo kung paano mo ipe-present ang content at kung paano gagawin ang layout.

## Ang Pagbabalik ng Art Direction sa Web Design

Siguro kanina mo pa napapansin na

paulit-ulit kong sinasabi itong “art direction”. Nawala kasi ang art sa Web design noong Responsive Web Design era. Pero sa Intrinsic Web Design, napakarami na nating tools na magagamit para sa kahit anong design at layout na maisip natin. Puwede na tayo ulit kumuha ng inspirasyon sa mga layout ng magazines, printed materials, at sa mga gawa ng mga artists na gaya ni Jan Tschichold.

Kasama sa mga layunin ng Intrinsic Web Design ay ang pagbabalik ng design sa Web. Kahit ano, puwede. Sa tulong ng mga bagong tools na built-in na sa CSS, makakagawa tayo ng eleganteng mga design. Kaya naman pinasisigla ang lahat, kapuwa mga designers at developers, na pag-aralan ang lahat ng options na mayroon para maging buhay muli ang Web, sa diwa na hindi na magkakamukha ang mga Web site.

Napakaraming halimbawa rito si Jen Simmons sa kaniyang [experimental lab](https://labs.jensimmons.com/) (<https://labs.jensimmons.com/>) (at inoobliga kitang tingnan ito para sa kapakanan mo 😊). Makikita mo sa Web

site niya ang iba’t ibang artworks sa tunay na mundo na ni-recreate niya gamit ang HTML at CSS. Pinatutunayan lang nito na marami tayong puwedeng gawin gamit ang mga tools na available na sa atin.

## **Ang Mga Tools ng Intrinsic Web Design**

Dahil nga ang pangako ng IWD ay built-in na sa CSS ang mga tools na gagamitin natin, at na hindi na natin kailangang gumamit ng mga “hack” o gumamit ng mga properties na ginawa para sa ibang mga bagay, dumami ang mga tools na magagamit natin.

- Flexbox
- Grid
- CSS Shapes
- Clip Path
- object-fit et. al
- Variable fonts
- @supports
- Bagong dagdag sa Writing Modes
- Viewport units (vh at vw)

Ang lahat ng ito ay built-in na sa CSS, at **hindi na natin kailangan ng frameworks**.

Ang article na ito ang una sa series ng mga articles tungkol sa Intrinsic Web Design. Sa mga susunod pang articles, susubukan nating talakayin ang iba't ibang tools na ito at kung paano sila gagamitin sa mga Web sites natin. Kaya laging i-check ang tag na [#intrinsic web design](#) sa Antares Blog.

## **Mga Tanong Tungkol sa Intrinsic Web Design**

### **Ibig bang sabihin nito, hindi na tayo gumagamit ng float?**

Siyempre hindi! Tandaan, ang float property ay ginawa na kagaya ng text wrapping sa Microsoft Word. May silbi pa rin ito. Sa katunayan, ang isang bagong feature ng CSS, ang CSS Shapes, ay nakasentro sa paggamit ng floated elements. At mismo sa site na ito, malawakang ginagamit sa tamang paraan ang mga float, gaya nitong [definition ng salitang “intrinsic”](#).

**Hindi natin puwedeng gamitin iyang bagong features ng CSS kasi may mga lumang browser pa rin na ginagamit sa mga kompanya, like iyong Internet Explorer.**

Hindi ito valid reason para iwasan ang mga bagong features ng CSS. Kung isasantabi mo ang pag-aaral ng mga bago sa CSS, inilalagay mo sa peligro ang career mo, dahil lahat ng iba pang developer inaaral na iyon. Ang CSS Grid ay supported na sa mahigit 88% ng mga Web users worldwide. At bukod pa ryan, mayroon tayong tinatawag na progressive enhancement. Kaya kung ang target users mo ay mga users ng IE at iba pang lumang browsers, puwede mong i-develop ang site mo gamit ang mga features na gumagana sa mga iyon, at saka mo idagdag ang mga bagong features na ginagamit na ng marami ngayon.

### **Eh ‘di say no to Bootstrap etc. na? Hindi na namin sila pag-aaralan?**

Nope, depende pa rin sa inyo iyon. Marami pa rin ang gumagamit ng

Bootstrap at Foundation, pati ang ibang mga framework. Kaya dapat alam pa rin natin kung paano sila gumagana para makahanap tayo ng trabaho. Ganoon naman lagi sa mundo ng technology; kahit na may bago nang lumabas, hangga't may gumagamit ng luma, kailangang alam natin kung paano gawin ang luma.

## **Paano kung ni-require kami na huwag gumamit ng mga bagong features ng CSS?**

Again, walang masama kung gagamitin natin iyong mga nakasanayan na natin. Kung kailangan mong mag-stick sa mga frameworks, walang problema. Kung gusto mong iwasan ang flexbox at CSS grid para gumamit ng float-based layout, nasa sa iyo ang desisyon. Ang mahalaga naman lagi rito ay kung ano ang magiging experience ng mga user ng site mo. At bukod pa diyan, ang experience ng mga developers sa paggawa ng Web sites. Mas madali kung tutuusin para sa atin na gamitin ang mga bagong tools na ito, pero wala rin namang masama kung gagamit tayo ng kutsilyo pambukas ng lata sa halip na gumamit ng can opener. Gagana iyon,

pero mas madali kung may can opener.

## **Conclusion**

Ano't anuman, nagbabago ang Web nang patuluyan. Para itong tren na patungo sa isang destinasyon: ang balanse sa pagitan ng developer experience at user experience. Bawat feature na nadadagdag sa Web platform ay para sa ikadadali ng buhay ng mga user, developer, at designers. Nasa sa iyo ang desisyon kung sasakay ka sa tren na ito. At isa pa, ang Intrinsic Web Design ay hindi isang bagong uso na malalaos pagkatapos ng ilang taon; at kahit malaos pa ito, ang mga technique nito ay nandito nang pangmatagalanan.

---

### **Footnotes**

1. ^ Wikipedia: [Tim Berners-Lee](https://en.wikipedia.org/wiki/Tim_Berners_Lee) ([https://en.wikipedia.org/wiki/Tim\\_Berners\\_lee](https://en.wikipedia.org/wiki/Tim_Berners_lee)).
2. Malibang binanggit sa mismong imahe, lahat ng illustrations sa article na ito ay kuha mula sa slides na ginawa ni Jen Simmons para sa An Event Apart 2018

## **Tingnan ang pinakabagong version ng artikulong ito sa**

<https://celestialcinnamon.github.io/antares-blog/tl/Intrinsic-Web-Design-Ito-na-ba-ang-Wakas-ng-Responsive-Web/>

### **See me outside**

- Antares on Facebook (<https://facebook.com/antaresprogramming>)
- Antares on Github (<https://github.com/celestialcinnamon/antares-blog>)
- See me on Facebook (<https://facebook.com/dorkas.rubio>)
- See me on Twitter (please don't)
- See me on Instagram (<https://instagram.com/melancholicapoptosis>)
- See my portfolio (<https://celestialcinnamon.github.io>)
- Send me an email: francoisoibur21@gmail.com

## **Ang Antares Programming**

Ang Antares Programming ay isang blog para sa mga Pilipino tungkol sa mga bagay tungkol sa Web at software development na hindi madalas maituro sa mga university at college. Dahil kinikilala ng Antares Programming ang epekto ng wikang kinalakhan o *mother tongue* sa pagkatuto, karamihan ng mga artikulo sa site na ito ay nasa Filipino. Umaasa ang writer nito na darating ang panahon na magkakaroon ng mas maraming materyal sa iba pang mga wika ng Filipinas. Pero sa ngayon, sapat na ang pagsisikap na ito.

Nga pala, hindi laging ganito "kalalim" (kapormal) ang Filipino sa site na ito. ☺

© 2019 Francis Rubio