

*presented by* **Alex Polybinsky & Sergey Zdobnov**  
*february, 2017*

# CSS-recipes

*key strategic tips*





# Basic notions

main approaches of web-development

## Progressive enhancement

- **Basic User Interaction for OLDER browsers**
- **Content and functional representation for OLDER browsers**
- *JS*
- *CSS3*

## Graceful degradation

- **JS**
- **CSS3**
- **User Interaction for MODERN browsers**
- **Content and functional representation for MODERN browsers**
- *Support for OLDER browsers*

## Mobile First

- **UI focusing on the most important aspects**
- **Simplicity and clarity of interface**
- *JS*
- *Support for OLDER browsers*

## Desktop First

- **Applicable for big portals with hundreds of items**
- *Support for OLDER browsers*

## Adaptive Web

- **One version of site**
- **Ability to update design for another devices**
- *Optimal utilization of all devices*
- *Continuity interface*

## Responsive Web

- **One version of layout**
- **Necessity of initial requirements for whole interface design**
- **Optimal utilization of all devices**
- **Continuity interface**



# Alignment

vertical and horizontal positioning

# Vertical Alignment



**vertical-align** and display: **inline-block** → alignment on **baseline**

New innovative processes  
created over industry best  
practices.

Global solutions for the new  
interconnected world.

# Vertical Alignment



## baseline and x-height





# Vertical Alignment



**vertical-align** and display: **table-cell** → alignment on **container**

New innovative processes  
created over industry best  
practices.

Global solutions for the new  
interconnected world.

# Useful links

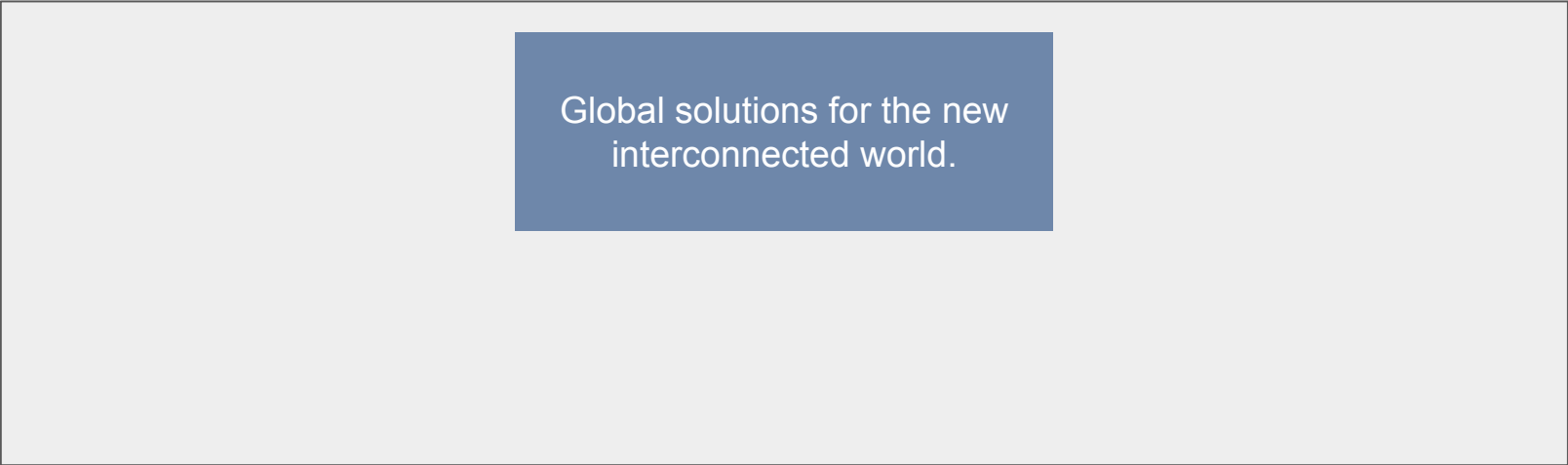


- ★ <https://developer.mozilla.org/en/docs/Web/CSS/vertical-align>
- ★ <http://css-tricks.com/what-is-vertical-align/>
- ★ <http://stackoverflow.com/questions/19366401/my-inline-block-elements-are-not-lining-up-properly>
- ★ <https://web-standards.ru/articles/vertical-align/>

# Horizontal Alignment



`text-align:center` for **inline** or **inline-block** → alignment on **container**



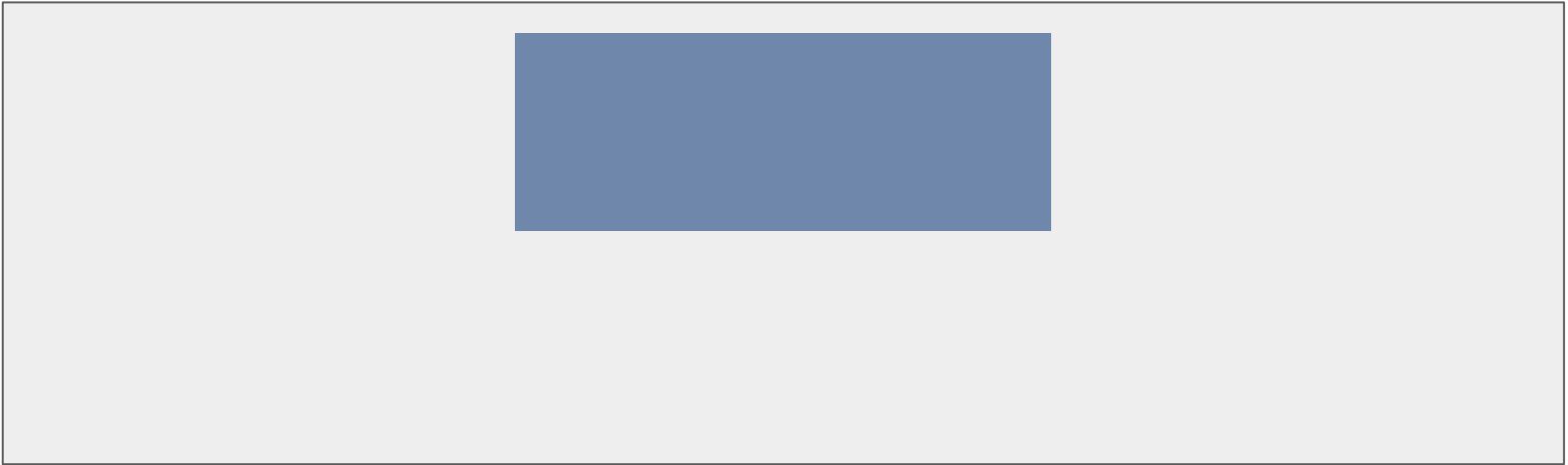
Global solutions for the new  
interconnected world.

The diagram shows a large light gray rectangular container. Inside this container, a smaller dark blue rectangular box is centered horizontally. The text "Global solutions for the new interconnected world." is centered within the blue box, demonstrating the effect of the `text-align:center` CSS property.

# Horizontal Alignment



margin: 0 auto; and display: **block**;



# Middle of Screen (1st way)



**strict** size, element's **position** (absolute)

**left** (50%), **top** (50%), **margin** (size/2)



# Middle of Screen (2nd way)



**strict** size, parent's **position** (relative), element's **position** (absolute)

**left** (0), **top** (0), **bottom** (0), **right** (0), **margin** (auto)

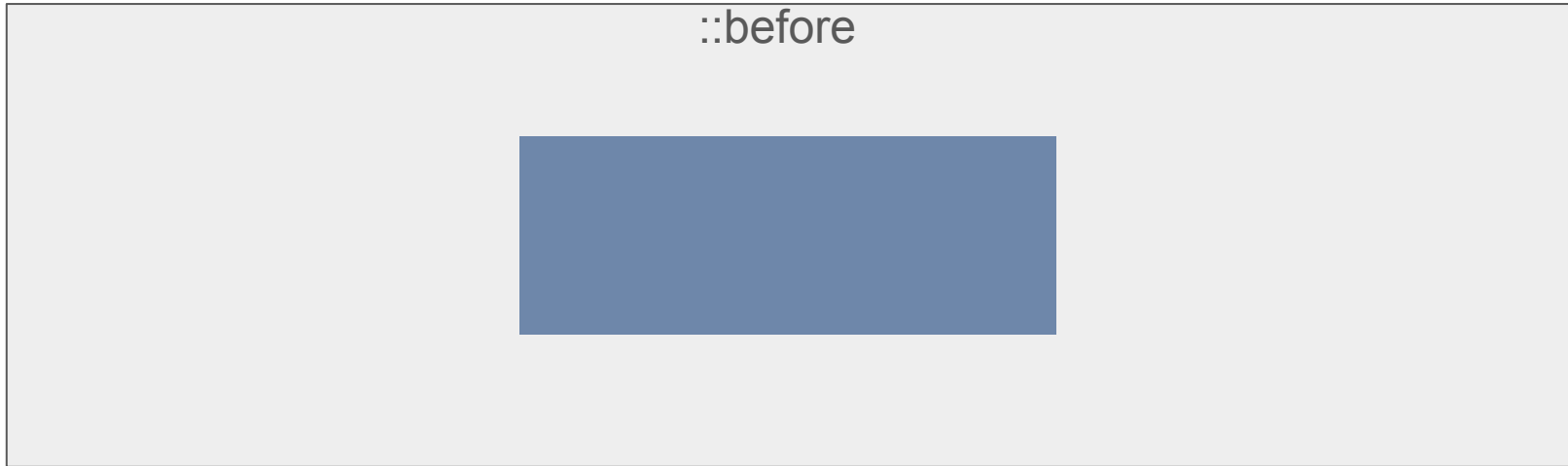


## Middle of Screen (3rd way)



**non-strict** size, parent's **position** (relative), element's **display** (inline-block)

**::before** contains **width** (1%), **height** (100%), **vertical-align** (middle)



## Middle of Screen (4th way)



**non-strict** size, parent's **position** (relative), element's **position** (absolute)

**left** (50%), **top** (50%), **transform: translate(-50%, -50%);**





# Useful links



- ★ <https://www.w3.org/Style/Examples/007/center.en.html>
- ★ <https://css-tricks.com/centering-css-complete-guide/>
- ★ <https://www.smashingmagazine.com/2013/08/absolute-horizontal-vertical-centering-css/>

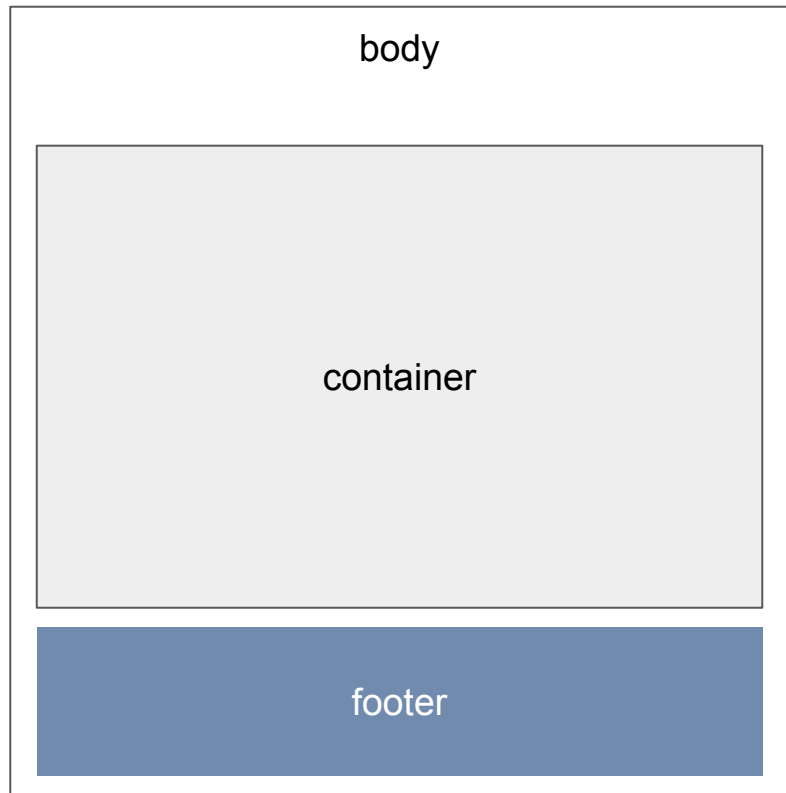
# Footer to bottom (1st way)

```
html { height: 100%; }
```

```
body { min-height: 100%; position: relative; }
```

```
container { padding-bottom: 70px; }
```

```
footer { position: absolute; height: 70px; left: 0; bottom: 0; }
```



# Footer to bottom (2nd way)

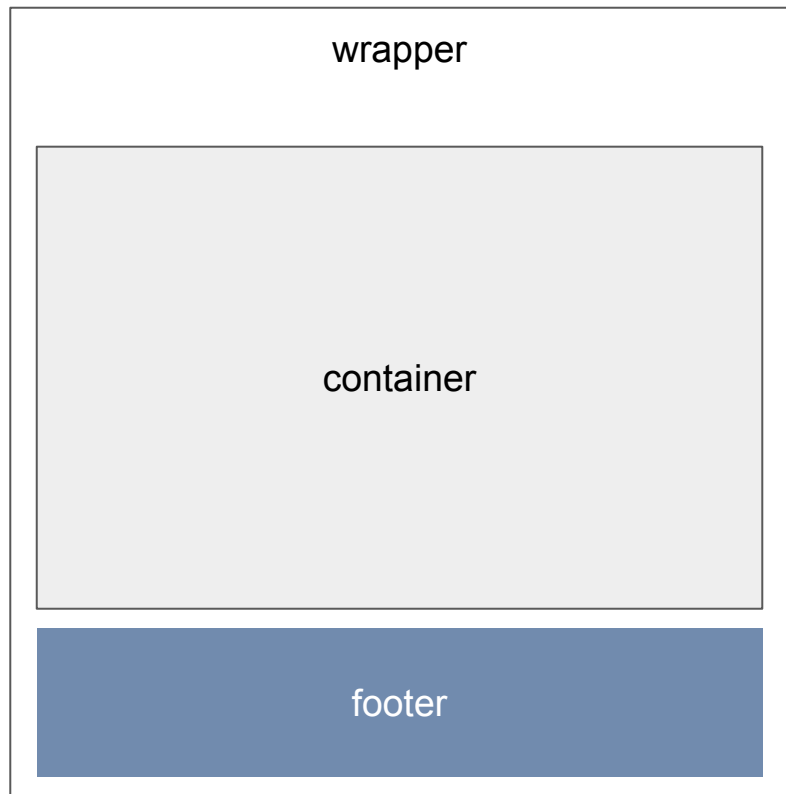


```
html { height: 100%; } body { height: 100%; }
```

```
wrapper { height: 100%; }
```

```
container { min-height: 100%;  
box-sizing: border-box; padding-bottom:  
70px; }
```

```
footer { box-sizing: border-box; height: 70px;  
margin-top: -70px; }
```

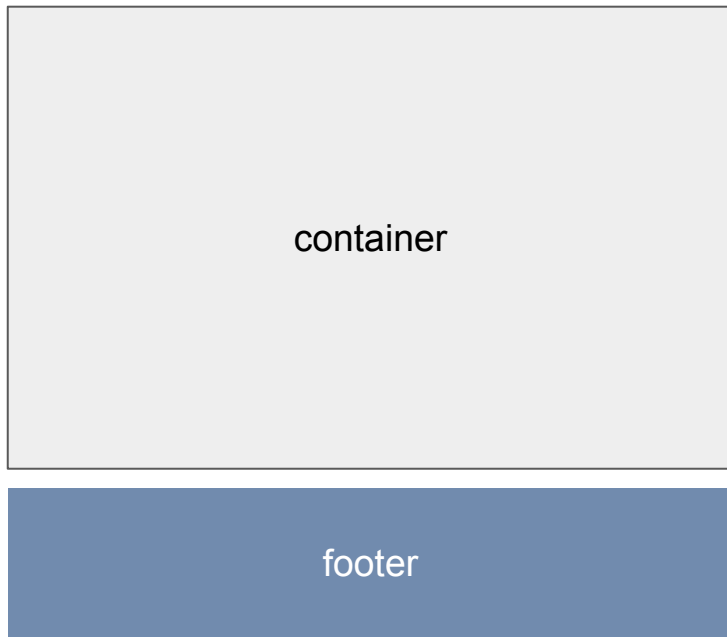


# Footer to bottom (3rd way)



```
container { calc(100vh - 70px);  
box-sizing: border-box; }
```

```
footer { box-sizing: border-box; height: 70px; }
```



# Useful links



- ★ [1st way example](#)
- ★ [2st way example](#)
- ★ [3st way example](#)

A black and white photograph of a young man and woman kissing in a crowd. The man is on the left, seen from the back, wearing a dark suit jacket. The woman is on the right, wearing a dark leather jacket, leaning in to kiss him on the cheek. They are surrounded by other people in a dimly lit setting, possibly a party or a social gathering. The background is dark and out of focus, with some people's faces visible.

# Popular recipes

necessary practical approaches

# Text Overflow



overflow (hidden), **text-overflow** (ellipsis), **white-space** (nowrap)

**block**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae massa sed arcu...

+ **table-layout** (fixed) for <table>

**td**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae massa sed arcu...

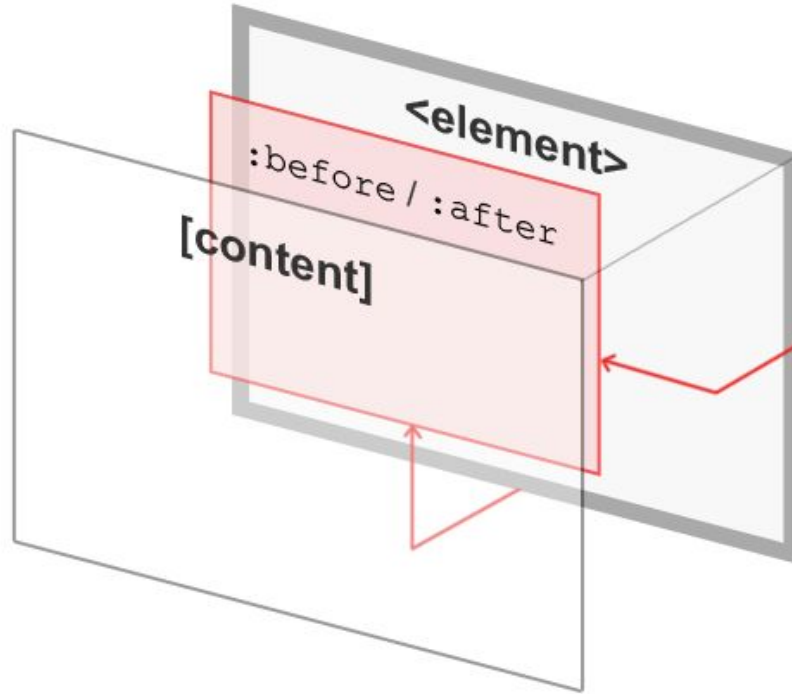
# Useful links



- ★ <https://css-tricks.com/snippets/css/truncate-string-with-ellipsis/>
- ★ <https://software.intel.com/en-us/html5/hub/blogs/ellipse-my-text/>
- ★ <http://dev.mobify.com/blog/multiline-ellipsis-in-pure-css/>



# Operations with `::before`, `::after`

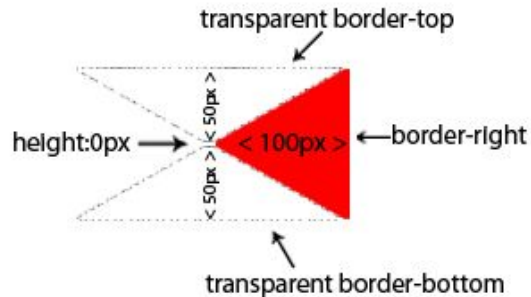


# Operations with ::before, ::after



```
<div class="example">Button with Up Arrow</div>
```

```
.example::before {  
  content: "";  
  display: block;  
  width: 0px;  
  height: 0px;  
  border-style: solid;  
  border-width: 0 100px 100px 100px;  
  border-color: transparent transparent red transparent;  
}
```





# Operations with ::before, ::after

```
<a class="tooltip" title="This is some information for our tooltip.">  
  <span>CSS3 Tooltip</span>  
</a>
```

```
.tooltip:hover::before {  
  triangle styles  
}
```

```
.tooltip:hover::after {  
  tooltip's box styles (background and font)  
}
```

[CSS3 Tooltip](#)

# Operations with ::before, ::after



## `.icon::before` { circle styles

```
content: "";  
position: absolute;  
top: 3px;  
left: 3px;  
width: 6px;  
height: 6px;  
border: 2px solid #ccc;  
border-radius: 14px;  
}
```

## `.icon::after` { line styles

```
content: "";  
position: absolute;  
top: 9px;  
left: 11px;  
width: 3px;  
height: 7px;  
margin-top: 0;  
background: #ccc;  
transform: rotate(-45deg);  
border-radius: 2px;  
}
```

# Operations with ::before, ::after



## `.icon::before` { line styles

```
content: "";  
display: block;  
height: 10px;  
width: 0;  
border-left: solid 2px #ccc;  
position: absolute;  
top: 4px;  
left: 4px;  
}
```

## `.icon::after` { triangle styles

```
content: "";  
height: 0;  
width: 0;  
border-style: solid;  
border-width: 5px 9px 5px 0;  
border-color: transparent #ccc transparent transparent;  
position: absolute;  
top: 4px;  
left: 8px;  
}
```

# Useful links



- ★ <https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements>
- ★ <https://css-tricks.com/almanac/selectors/a/after-and-before/>
- ★ <https://css-tricks.com/pseudo-element-roundup/>
- ★ <https://www.amazeelabs.com/en/blog/three-things-you-can-do-with-css-pseudo-elements>
- ★ <https://www.smashingmagazine.com/2011/03/styling-elements-with-glyphs-sprites-and-pseudo-elements/>
- ★ [TOOLTIP](#)
- ★ [ICONS](#)

# Customization (input + label)



```
<input id="inputId" />  
<label for="inputId">Custom style</label>
```

```
input {  
  display: none;  
}
```

```
input + label {  
  custom input styles (inactive)  
}
```

```
input:checked + label {  
  custom input styles (active)  
}
```

## Checkboxes

- My first checkbox
- The second checkbox
- Another checkbox

## Radios

- A radio
- Another radio
- One more radio

# Useful links



★ <https://css-tricks.com/float-labels-css/>

★ <http://htmlbook.ru/blog/kartinka-vmesto-chekboksa>





# Tables

options of table layout

# Table Layout



**table-layout: auto** (*default*)

Column 1	Column 2	Column 3
This is your text.	This is yooooooooooooooooooooooooooooooooour text.	This is your text.

**table-layout: auto + width: 100%**

Column 1	Column 2	Column 3
This is your text.	This is yooooooooooooooooooooooooooooooooour text.	This is your text.

# Table Layout



**table-layout: fixed**

Column 1	Column 2	Column 3
This is your text.	This is yooooooooooooooooooooooooooooour text.	This is your text.

**table-layout: fixed + width: 100%**

Column 1	Column 2	Column 3
This is your text.	This is yooooooooooooooooooooooooooooour text.	This is your text.

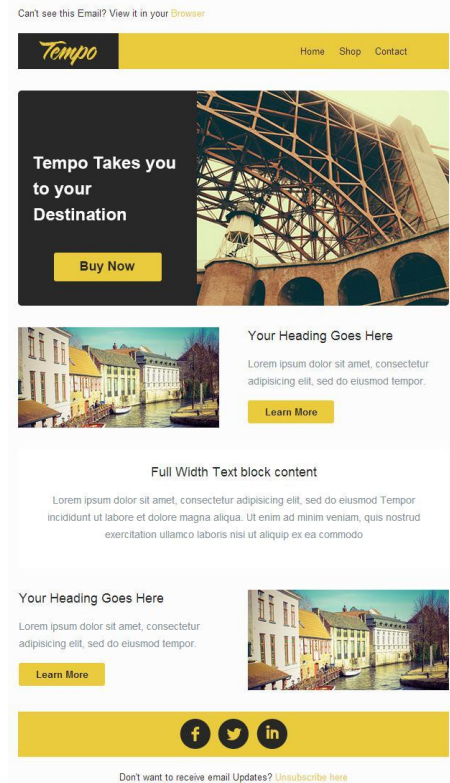
# Useful links



- ★ [Scrolling in table](#)
- ★ <https://www.w3.org/wiki/CSS/Properties/table-layout>
- ★ <https://css-tricks.com/almanac/properties/t/text-overflow/>

# Emails\*

- ❖ **Table** structure of frame and content items (`<table>` *inside* `<table>`)
- ❖ **Fixed** size of main container
- ❖ Use and support of **obsolete** properties and attributes (**align**, **bg-color**, **width**)
- ❖ Possibility of **adaptive** layout



# Useful links



- ★ [Example #1](#)
- ★ [Adaptive email-letters](#)
- ★ <https://litmus.com/community/templates>



**GOOD NIGHT** *and*  
**GOOD LUCK**

---