Bootstrap 5

Introduction and Layout

What is Bootstrap

- The most popular HTML, CSS, and JS framework for developing responsive, mobile-first projects on the web.
- It’s a frontend development framework with giant collection of handy, reusable bits of code written in HTML, CSS, and JavaScript that enables developers and designers to quickly build fully responsive websites.
- Bootstrap was originally created by a designer and a developer at Twitter in mid-2010. Before being an open-sourced framework, Bootstrap was known as Twitter Blueprint.

Why to use Bootstrap

- **Save lots of time** — You can save lots of time and efforts using the Bootstrap predefined design templates and classes and concentrate on other development work.
- **Responsive features** — Can easily create responsive websites that appear more appropriately on different devices and screen resolutions without any change in markup.
- **Consistent design** — All Bootstrap components share the same design templates and styles through a central library, so the design and layout of your web pages will be consistent.
- **Easy to use** — Anybody with the basic working knowledge of HTML, CSS and JavaScript can start development with Bootstrap.
- **Compatible with browsers** — Bootstrap is created with modern web browsers in mind and it is compatible with all modern browsers such as Chrome, Firefox, Safari, Internet Explorer, etc.
- **Open Source** — And the best part is, it is completely free to download and use.

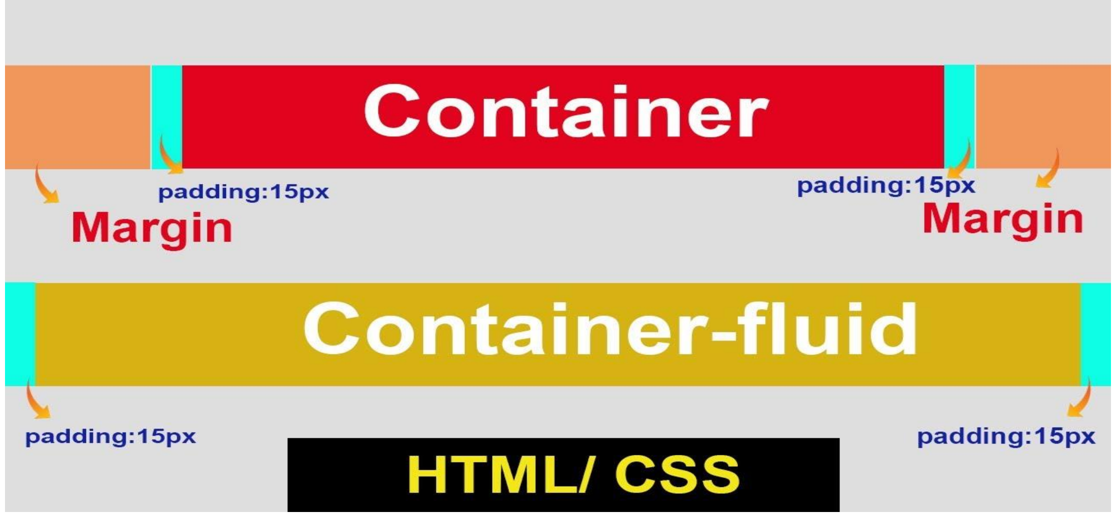
Breakpoints

- Breakpoints are customizable widths that determine how your responsive layout behaves across device or viewport sizes in Bootstrap.
- Bootstrap includes six default breakpoints, sometimes referred to as grid tiers, for building responsively. These breakpoints can be customized if you’re using our source Sass files.

x-small	**none**	**<576px**
small	sm	≥576px
medium	md	≥768px
large	lg	≥992px
Extra large	xl	≥1200px
Extra Extra Large	xxl	≥1400px

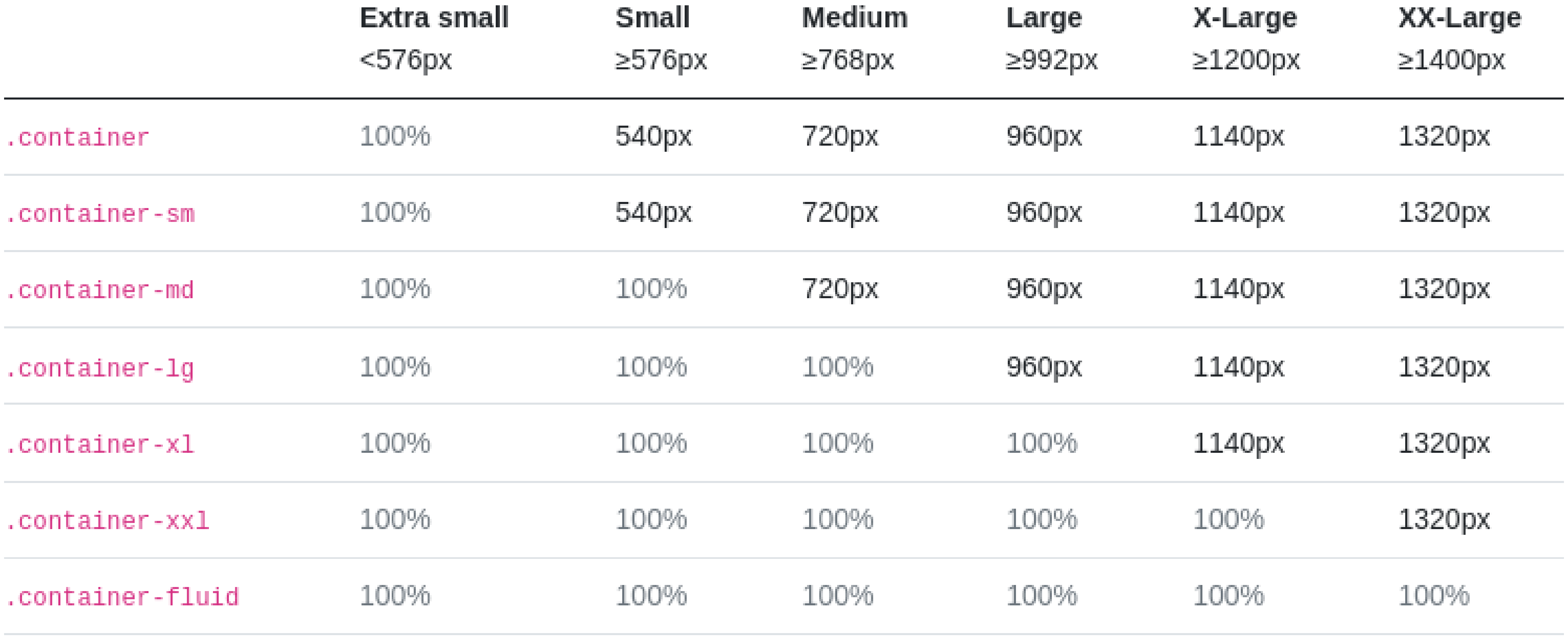
Containers

- Containers are a fundamental building block of Bootstrap that contain, pad, and align your content within a given device or viewport.
- The .container class provides a responsive fixed width container.
- The .container-fluid class provides a full width container, spanning the entire width of the viewport.



container-{breakpoint}

Use .container-fluid when you want your page to shapeshift with every little difference in its viewport size.Use .container when you want your page to shapeshift to only 5 kinds of sizes, i.e. "breakpoints".

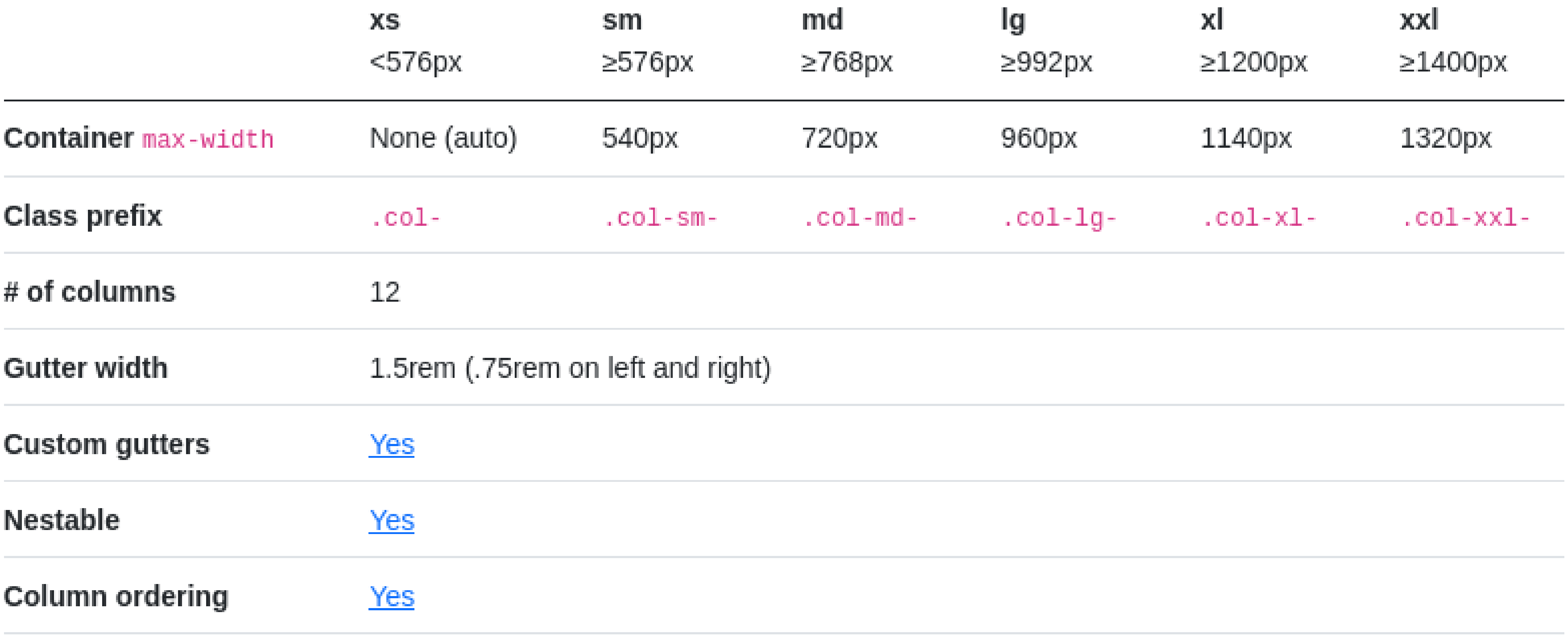


	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
`.container`	100%	540px	720px	960px	1140px	1320px
`.container-sm`	100%	540px	720px	960px	1140px	1320px
`.container-md`	100%	100%	720px	960px	1140px	1320px
`.container-lg`	100%	100%	100%	960px	1140px	1320px
`.container-xl`	100%	100%	100%	100%	1140px	1320px
`.container-xxl`	100%	100%	100%	100%	100%	1320px
`.container-fluid`	100%	100%	100%	100%	100%	100%

Grid

- Bootstrap’s grid system uses a series of containers, rows, and columns to layout and align content. It’s built with flexbox and is fully responsive.
- It allows up to 12 columns across the page for a row. More than that, columns will stack no matter the viewport.
- The columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items were stacked on top of each other.

Grid options



	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container `max-width`	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`	`.col-xxl-`
# of columns	12					
Gutter width	1.5rem (.75rem on left and right)					
Custom gutters	Yes					
Nestable	Yes					
Column ordering	Yes					

Auto Layout Columns

Equal Width

Create equal-width columns across all devices and viewports using .row and .col grid classes.

Setting one column width

Auto-layout for flexbox grid columns also means you can set the width of one column and have the sibling columns automatically resize around it.

Variable width content

Use col-{breakpoint}-auto classes to size columns based on the natural width of their content.

Exercises

- Create a grid with 2 rows and 4 equal size columns
- Create a grid with 2 rows with middle column width of 4 in 1st row and last column width of 5 columns in second row
- Create a variable width column grid with first column having 2x content than 2nd column and 2nd column having 3x content than 3rd column.

Responsive Classes

Customize the size of columns on extra small, small, medium, large, or extra large devices.

For grids that are the same from the smallest of devices to the largest, use the .col and .col-* classes. Specify a numbered class when you need a particularly sized column; otherwise, feel free to stick to .col.

Stacked to Horizontal

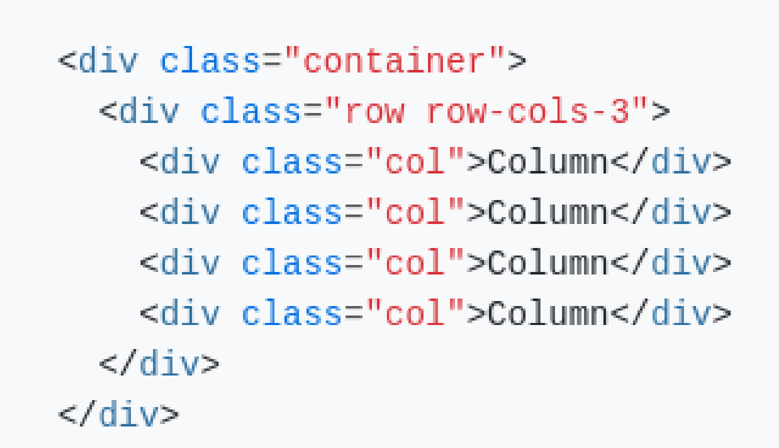
Using a single set of .col-sm-* classes, you can create a basic grid system that starts out stacked and becomes horizontal at the small breakpoint (sm).

Mix and match

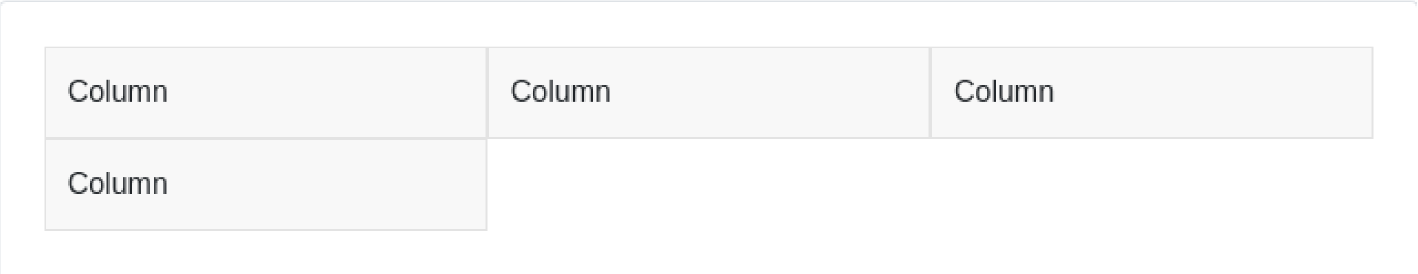
Don’t want your columns to simply stack in some grid tiers? Use a combination of different classes for each tier as needed.

Row Columns

Use the responsive .row-cols-* classes to quickly set the number of columns that best render your content and layout. Whereas normal .col-* classes apply to the individual columns (e.g., .col-md-4), the row columns classes are set on the parent .row as a shortcut. With .row-cols-auto you can give the columns their natural width.



```
<div class="container">
  <div class="row row-cols-3">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```



Column	Column	Column
Column		

Exercises

- Create a Grid that has 4 columns in large screens, 2 columns in medium screens.
- Create a Grid using row-col* classes with 4 columns .

Columns

Align, order, and offset columns using classes.

Alignment

Use flexbox alignment utilities to vertically and horizontally align columns.

Vertical Alignment

All columns in a row - .align-items-start, .align-items-center, .align-items-end

Each column in a row - .align-self-start, .align-self-center, .align-self-end

Horizontal Alignment

.justify-content-start, .justify-content-center, .justify-content-end, .justify-space-around, .justify-content-space-between, .justify-content-space-evenly

Column Wrapping & Break

- If more than 12 columns are present in a row, columns wrap.

- To force a break the columns into new row

```
<div class="w-100"></div>
```

Ordering Columns

- Use .order- classes for controlling the visual order of your content.
- Offsetting columns
- You can offset grid columns in two ways: our responsive .offset- grid classes and our margin utilities. Grid classes are sized to match columns while margins are more useful for quick layouts where the width of the offset is variable.

Exerciess

- Create a 3x3 grid, first row should sit in the end, second row should be centered vertically, 3rd row columns should be equally placed
- Create a grid give space with offset in the first row and use 10% margin in the second row , second column and third column display should be toggled.

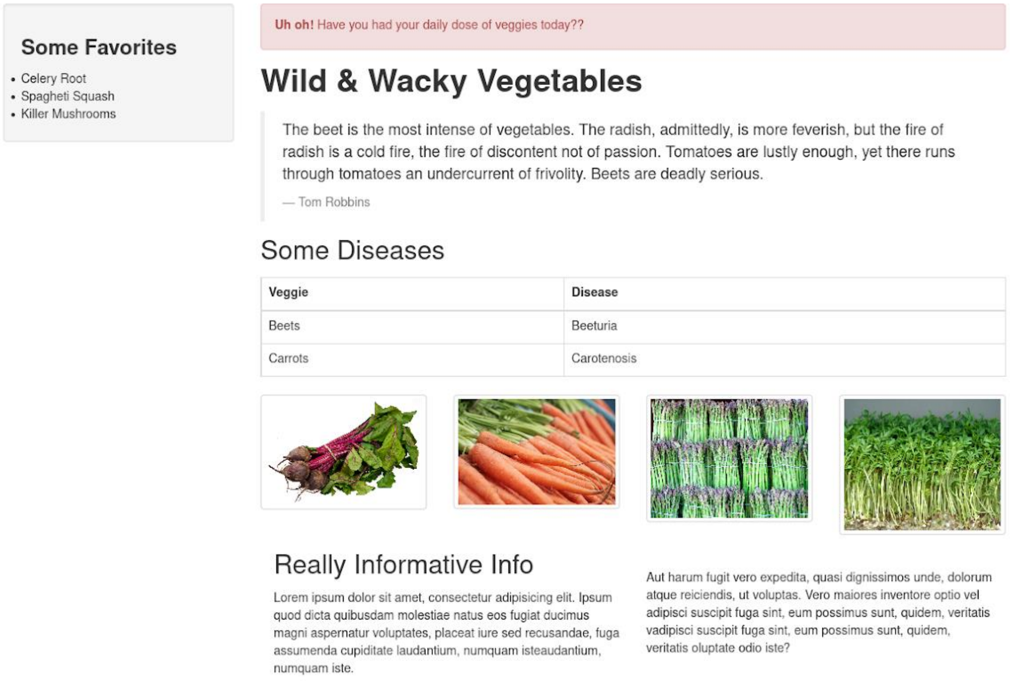
Exercises

- Create 5x5 grid with 2 space n small screens and 3x space in larger screens.Horizontal space is 2, vertical space is 3.

Readings

- Grid system · Bootstrap v5.1 (getbootstrap.com)
- https://getbootstrap.com/docs/5.1/layout/columns/

Assignment



Some Favorites

- Celery Root
- Spagheti Squash
- Killer Mushrooms

Uh oh! Have you had your daily dose of veggies today??

Wild & Wacky Vegetables

> The beet is the most intense of vegetables. The radish, admittedly, is more feverish, but the fire of radish is a cold fire, the fire of discontent not of passion. Tomatoes are lustly enough, yet there runs through tomatoes an undercurrent of frivolity. Beets are deadly serious.
>
> — Tom Robbins

Some Diseases

Veggie	Disease
Beets	Beeturia
Carrots	Carotenosis

Really Informative Info

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ipsum quod dicta quibusdam molestiae natus eos fugiat ducimus magni aspernatur voluptates, placeat iure sed recusandae, fuga assumenda cupiditate laudantium, numquam isteaudantium, numquam iste.

Aut harum fugit vero expedita, quasi dignissimos unde, dolorum atque reiciendis, ut voluptas. Vero maiores inventore optio vel adipisci suscipit fuga sint, eum possimus sunt, quidem, veritatis vadipisci suscipit fuga sint, eum possimus sunt, quidem, veritatis oluptate odio iste?