CSS Grid Project

This project consists of three parts. You will submit each part separately. Each part will be graded separately.

Part 1: Wireframe a responsive grid layout

Before You Begin:

1. Examine the three screenshots in the **Assets/Part 1** folder.
2. You will code a single CSS file to replicate the responsive page using CSS Grid. (A starter HTML file and a CSS file with basic styling are provided. Don't edit either of these.)
3. Create a folder named **last-first-grids-1**.
4. Copy both the *starter-file.html* and *formatting.css* files to your folder.
5. Rename the copied HTML file to something appropriate for a home page.
6. Organize your CSS file in a subfolder named **css**.
7. Double-click the HTML file. It should open in your browser with the basic CSS styling applied (colored rows).

Restrictions:

1. You must use numbered grid lines and not grid-area for Part 1.
2. You may not edit any HTML at all in Part 1.
3. Your CSS must address "desktop first" design (as opposed to mobile-first).
4. All code you write must adhere to our Code Formatting Guidelines.

Tips:

1. Study the screenshots in the *Assets/Part 1* folder so you are clear on what you must replicate.
2. Sketch your grid on paper first so you are clear on how many rows/columns are needed.
3. Study the HTML code before writing any CSS.
4. Once you start coding, use Firefox's Developer Tools and enable the layout grid so you can see the grid and lines.
5. Rewatch the *CSS: Advanced Layouts with Grid* course on Lynda.com. Specifically *Chapters 1-3*.

To Do:

1. Create a new CSS file named **layout.css** and store it in the **css** subfolder.
2. Code the default layout first – the one to display the elements on a wide computer screen as shown in *01-Desktop.jpg* (in the *Assets/Part 1* folder).
3. Create a media query breakpoint for 780px and narrower that displays the elements as shown in *02-780.jpg*.
4. Create a media query breakpoint for 400px and narrower that displays the elements as shown in *03-400.jpg*.
5. Validate your CSS and ensure it passes validation.
6. Test your page in Firefox, Chrome, Edge (Windows users), and Safari (Mac users) and ensure it is fully responsive in each.

Submission:

1. Zip your **last-first-grids-1** folder and submit via Moodle.

Part 2: Wireframe a nested grid

Before You Begin:

1. Examine the three screenshots in the *Assets/Part 2* folder.
2. You will code a single CSS file to replicate the layout using nested CSS Grids. (A starter HTML file and a CSS file with basic styling are provided. Don't edit either of these.)
3. Create a folder named **last-first-grids-2**.
4. Copy both the starter-file2.html and formatting2.css files to your folder.
5. Rename the copied HTML file to something appropriate for a home page.
6. Organize your CSS file in a subfolder named **css**.
7. Double-click the HTML file. It should open in your browser with the basic CSS styling applied (colored rows).

Restrictions:

1. You must use numbered grid lines and not grid-area for Part 2.
2. You may not edit any HTML at all in Part 2.
3. All code you write must adhere to our Code Formatting Guidelines.

Tips:

1. Study the screenshots in the *Assets/Part 2* folder so you are clear on what you must replicate.
2. Sketch your grid on paper first so you are clear on how many rows/columns are needed.
3. Study the HTML code before writing any CSS.
4. Once you start coding, use Firefox's Developer Tools and enable the layout grid so you can see the grid and lines.

To Do:

1. Create a new CSS file named **layout2.css** and store it in the **css** subfolder.
2. In the CSS, add a grid to the MAIN element, giving it 3 columns (make the center column twice as wide as the other two) and 2 rows (auto-sized).
3. Preview your page in a browser. It should match the *Assets/Part 2/02-main-grid.jpg* screenshot.
4. Code the *main content C* block to display in the 1st column of the top row, as in *Assets/Part 2/03-main-c.jpg*. The subcontent (the entire <section> element) now displays in column 1 of the second row. You can see this if you are visualizing your grid with Firefox's Developer Tools.
5. Make the <section> span all three columns of the second row, as in *Assets/Part 2/04-section.jpg.*
6. To create a nested grid to control the layout of the children <section>, make <section> display as a grid with 2 equally sized columns and 1 auto-sized row. Do this in the same rule as above – the one that makes the <section> span 3 columns. It should match *Assets/Part 2/05-section-nested-grid.jpg.*
7. Make the *subcontent A* block display in the right column of the grid's single row. Your page should match *Assets/Part 2/06-complete.jpg*
8. Validate your CSS and ensure it passes validation.
9. Test your page in Firefox, Chrome, Edge (Windows users), and Safari (Mac users) and ensure it displays as expected.

Submission:

1. Zip your **last-first-grids-2** folder and submit via Moodle.

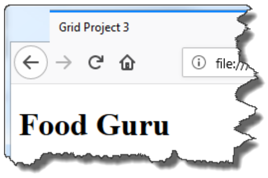
Part 3: Apply your grid knowledge to an actual design (responsive nested grids)

Before You Begin:

1. You will code all of the HTML and CSS for this part.
2. I will walk you through the HTML coding step by step.
3. I will walk you through the basic CSS styling step by step.
4. You will need to code the responsive nested grid layout completely on your own.
5. Create a folder named **last-first-grids-3**.
6. Create subfolders named **css** and **images**.
7. Copy the images from the *Assets/Part 3/Resources* folder to your images folder.
8. All of text you need to type can be copied from the *Assets/Part 3/Resources/text.txt* file.

Code the HTML

1. In the *last-first-grids-3* folder, create an HTML file named **index.html**.
2. Code the basic HTML skeleton – doctype, head (with meta/charset and title), and an empty body.
3. The remainder of the code should be inside the <body> in the order presented.
4. Code a <header> with an <h1> and the text Food Guru.

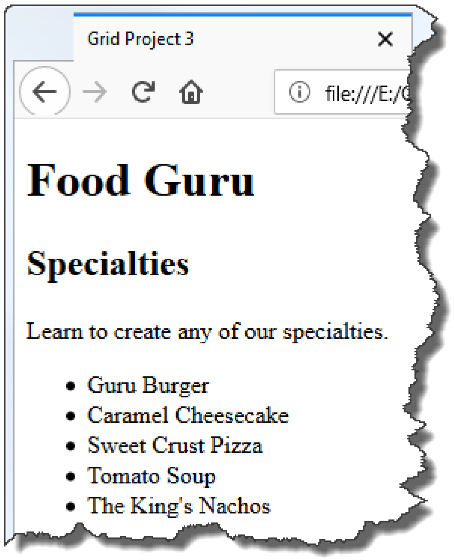


Food Guru

5. Code a <main> block.

Specialties

6. Within <main>, code a <div> with an ID of *specialties* that encloses an <h2>, <p>, and <ul> as shown:



Food Guru

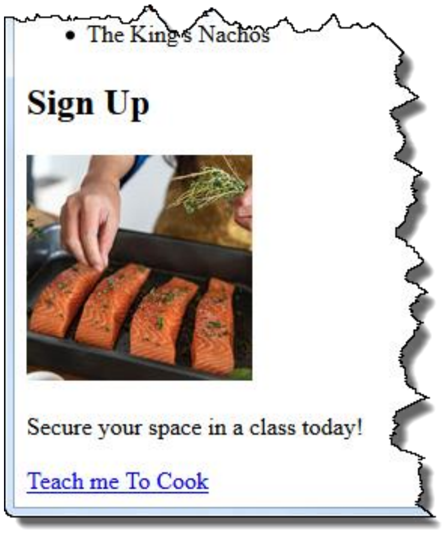
Specialties

Learn to create any of our specialties.

- Guru Burger
- Caramel Cheesecake
- Sweet Crust Pizza
- Tomato Soup
- The King's Nachos

Signup

7. Code a second <div>, with an ID of *signup*, to enclose an <h2>, the signup image, <p>, and <a> as shown. (Set the anchor to a null link: `href="#"`.)



• The King's Nachos

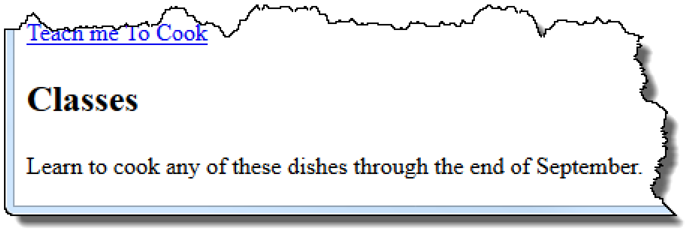
Sign Up

Secure your space in a class today!

Teach me To Cook

Classes

8. Still within <main>, code a <section> element with an ID of *classes* to hold the introduction and blurbs for 8 classes. (This <section> will house 9 children.)
9. Inside the <section>, code a <header> that encloses an <h2> and <p> as shown:

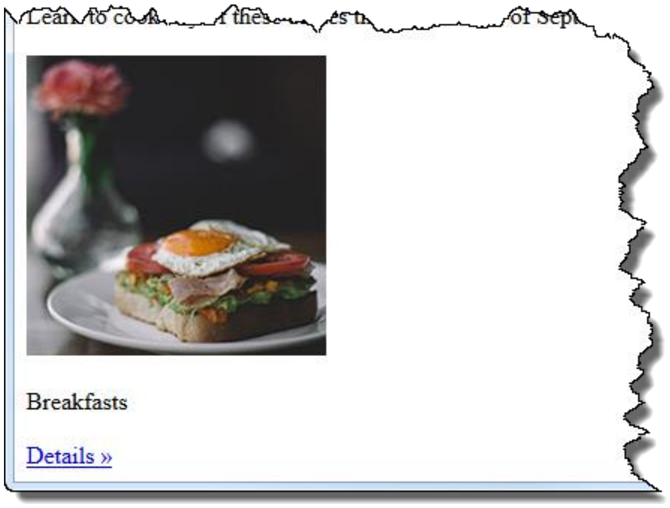


Teach me To Cook

Classes

Learn to cook any of these dishes through the end of September.

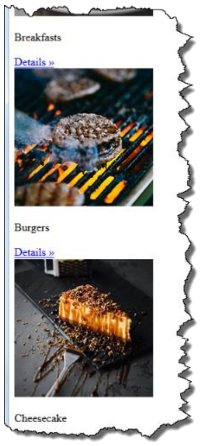
10. Still inside <section id="classes"> and after the <header>, code a <div> with no ID that encloses the breakfast image, <p>, and null link as shown. Use the character `&raquo;` to create the arrows in the link.



Breakfasts

Details »

11. Still inside <section id="classes">, code 7 more <div> elements just like the previous one to enclose additional sets of image, paragraph, and null link. Use the following images: *burger, cheesecake, cookies, sharing, pizza, sandwich, soup*
    You should have 8 of these blocks stacked on top of each other, along with the <header> all inside <section id="classes">



Breakfasts

Details »

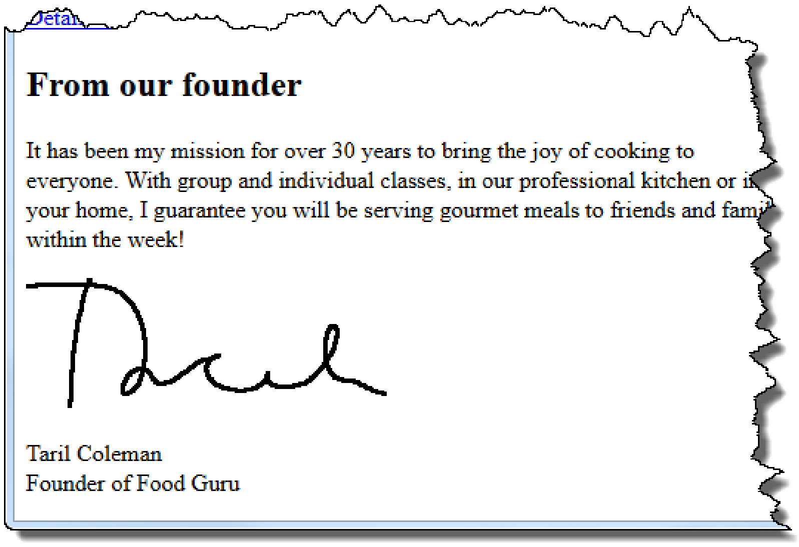
Burgers

Details »

Cheesecake

Founder

12. Code a new <section> with an ID of *founder* as a sibling to <section id="classes">.
13. Inside <section id="founder">, code two children:
    a. A <div> with an <h2>, paragraph, signature image, and another paragraph with a line break. The paragraph text can be copied from: *Assets/Part 3/Resources/text.txt*



Detai

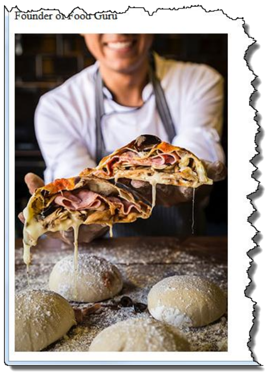
From our founder

It has been my mission for over 30 years to bring the joy of cooking to everyone. With group and individual classes, in our professional kitchen or in your home, I guarantee you will be serving gourmet meals to friends and fami within the week!

[illegible]

Taril Coleman
Founder of Food Guru

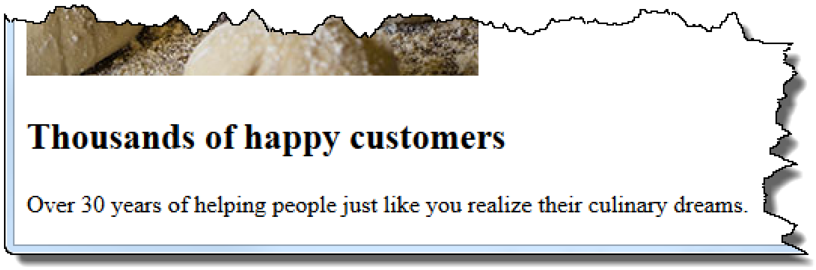
    b. The *founder* image.



Founder of Food Guru

Testimonials

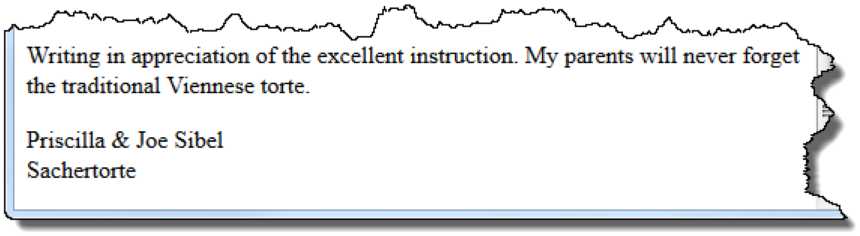
14. Code a new <section> with an ID of *testimonials* as a sibling to <section id="founder">.
15. Within <section id="testimonials">, code a <header> to hold the introduction of an <h2> and a paragraph:



Thousands of happy customers

Over 30 years of helping people just like you realize their culinary dreams.

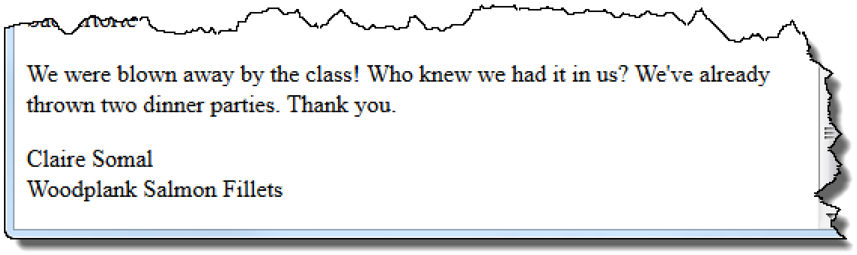
16. Still within <section id="testimonials"> and after the <header>, code a <div> to hold the following two paragraphs – the second of which has a line break:



Writing in appreciation of the excellent instruction. My parents will never forget the traditional Viennese torte.

Priscilla & Joe Sibel
Sachertorte

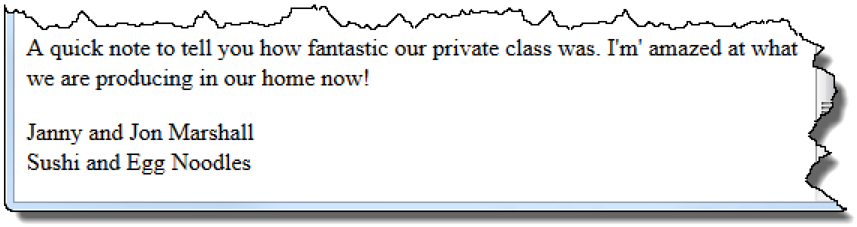
17. Code a second testimonial, also a <div>, holding two similar paragraphs:



We were blown away by the class! Who knew we had it in us? We've already thrown two dinner parties. Thank you.

Claire Somal
Woodplank Salmon Fillets

18. Code the final testimonial:



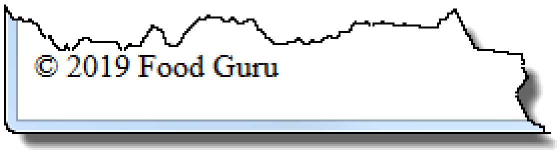
A quick note to tell you how fantastic our private class was. I'm' amazed at what we are producing in our home now!

Janny and Jon Marshall
Sushi and Egg Noodles

19. That concludes the <main> block.

Footer

20. Code a <footer> with a single paragraph:



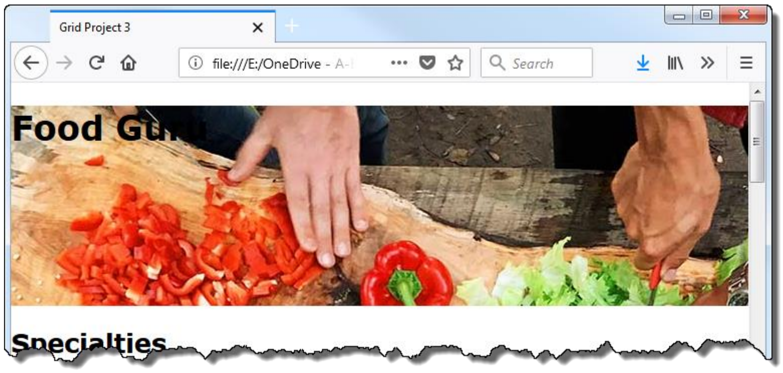
© 2019 Food Guru

Test

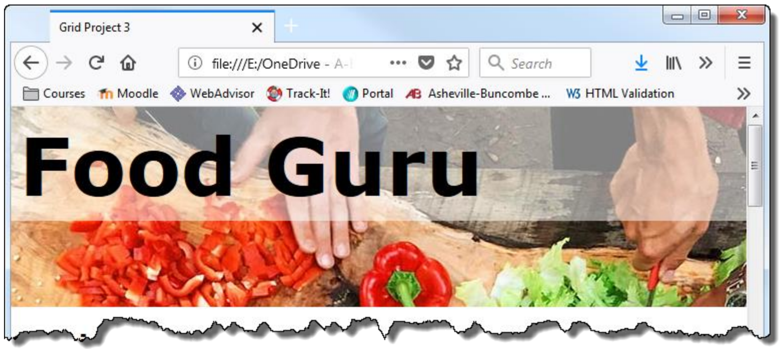
21. Validate your code and ensure there are no errors.
22. Test your page in the browser and ensure it matches *Assets/Part 3/Comps/01-HTML-only.jpg*. Note your words will wrap differently based on browser width.

Code the basic CSS styling

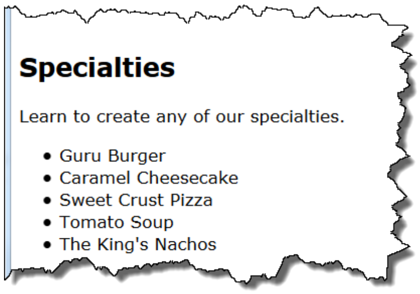
1. In your HTML file, link to a CSS file named styling3.css, which should be stored in the *css* folder.
2. In your HTML, add the <meta> tag required for media queries to work for mobile devices and responsive sites.
3. Create the *styling3.css* file in the *css* folder.
4. Code the following in the order presented. Each step should comprise a single CSS rule.
5. Target the *universal selector* to set the box-sizing to use border-box.
6. Target the body to use the verdana typeface (arial and sans-serif as alternatives) as a size of 17px. Use font shorthand. Remove the margin on all sides.
7. Use a child selector to target the single header that is a child of the body. Set its background-image to use banner.jpg. Set the bottom margin to 1rem and the minimum height to 200px.



8. Set the H1 in only this HEADER to have a background color of white, faded to 50%. Set the font size to 5rem. Remove all its margins and give it padding of .5rem on all sides.



9. Add .5rem of padding to the #specialties element.

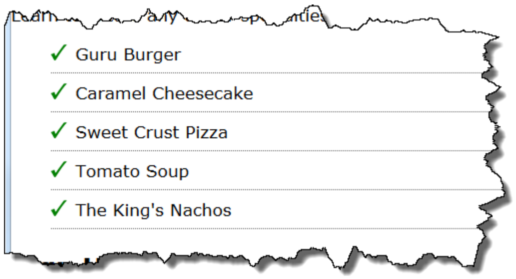


Specialties

Learn to create any of our specialties.

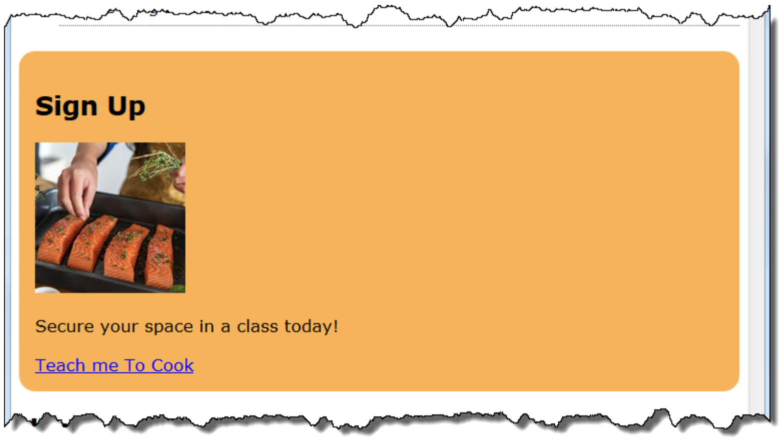
- Guru Burger
- Caramel Cheesecake
- Sweet Crust Pizza
- Tomato Soup
- The King's Nachos

10. Target all list items inside of #specialties and set them to use checkmark.png as the list image. Add a bottom border of 1px dotted and colored #666. Set the list style's position so that the bottom border extends under the list image. Set both the bottom margin and bottom padding to .5rem.



✓ Guru Burger
✓ Caramel Cheesecake
✓ Sweet Crust Pizza
✓ Tomato Soup
✓ The King's Nachos

11. Target the #signup element to use a background color of #f5b45b, a border radius of 1rem, padding on all sides of 1rem, top/bottom margin of 0, and left/right margin of .5rem.

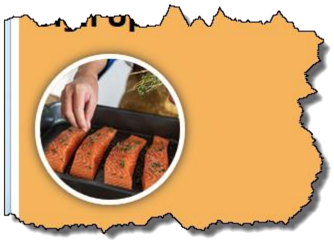


Sign Up

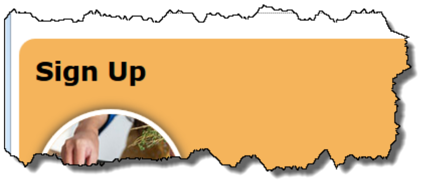
Secure your space in a class today!

Teach me To Cook

12. Style the image inside of #signup to have a 5px solid white border, a border radius of 50%, and a box shadow colored #333 with no offsets and a blur of 10px:

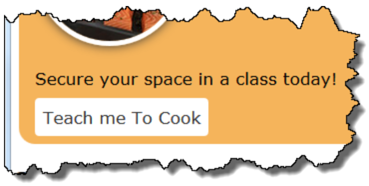


13. Remove the top margin of the heading inside of #signup.



Sign Up

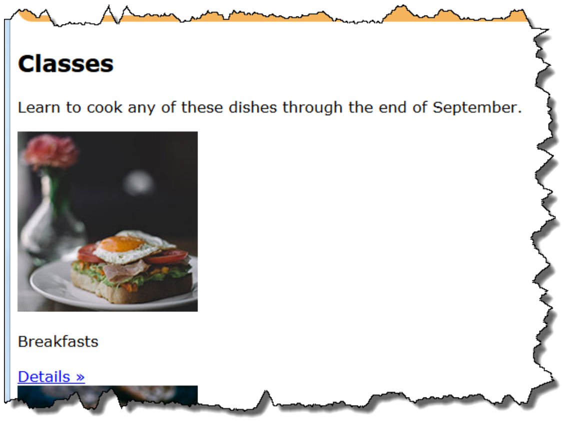
14. Style the link inside of #signup to have a white background, border radius of .25rem, and padding of .5rem. Remove its underline and color the link #333.



Secure your space in a class today!

Teach me To Cook

15. Give the #classes element .5rem of padding on all sides.



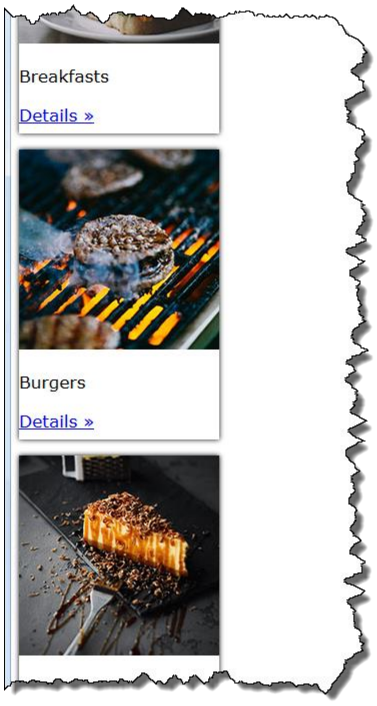
Classes

Learn to cook any of these dishes through the end of September.

Breakfasts

Details »

16. Target all the <div> elements inside #classes to have a box shadow colored black with a blur of 5px and no offsets. Set their maximum width to 200px, add a bottom margin of 1rem and bottom padding of .5rem.



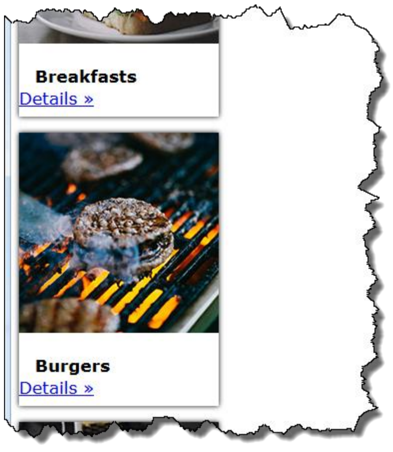
Breakfasts

Details »

Burgers

Details »

17. Style the paragraphs inside of the <div> elements within #classes (but not inside the <header>) to be bold with no bottom margin and left padding of 1rem:



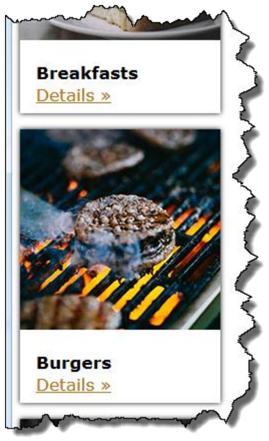
Breakfasts

Details »

Burgers

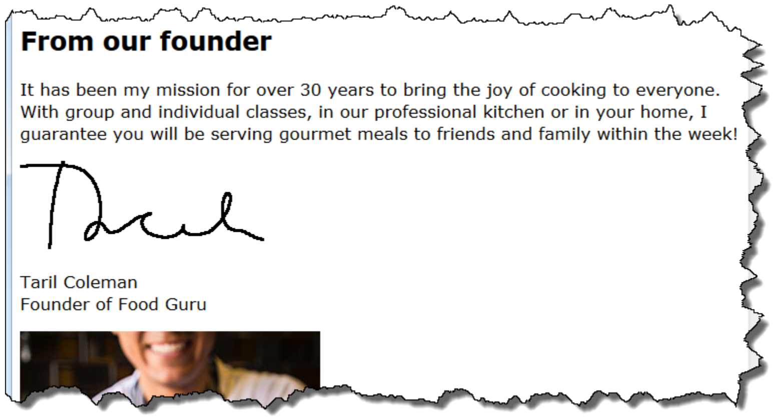
Details »

18. Style the links inside of the <div> elements within #classes (but not inside the <header>) to be colored #af760e with left padding of 1rem:


Breakfasts
Details »

Burgers
Details »

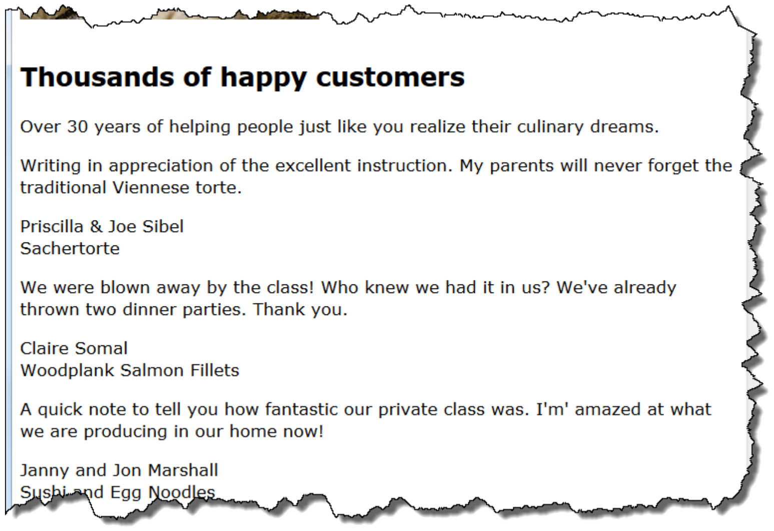
19. Add .5rem of padding to the #founder element.


From our founder

It has been my mission for over 30 years to bring the joy of cooking to everyone. With group and individual classes, in our professional kitchen or in your home, I guarantee you will be serving gourmet meals to friends and family within the week!

Taril Coleman
Founder of Food Guru

20. Add .5rem of padding to the #testimonials element.


Thousands of happy customers

Over 30 years of helping people just like you realize their culinary dreams.

Writing in appreciation of the excellent instruction. My parents will never forget the traditional Viennese torte.

Priscilla & Joe Sibel
Sachertorte

We were blown away by the class! Who knew we had it in us? We've already thrown two dinner parties. Thank you.

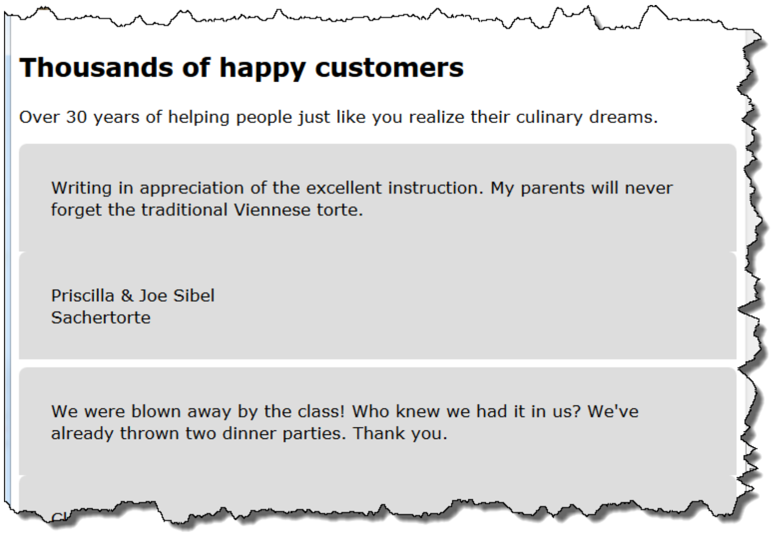
Claire Somal
Woodplank Salmon Fillets

A quick note to tell you how fantastic our private class was. I'm' amazed at what we are producing in our home now!

Janny and Jon Marshall
Sushi and Egg Noodles

21. Style all the <div> elements inside of #testimonials to have a bottom margin of .5rem.

22. Style the paragraphs inside the <div> elements – and not in the <header> – inside of #testimonials to have a background color of #ddd, padding of 2rem, and no margin. Round only the top two corners to .5rem.



Thousands of happy customers

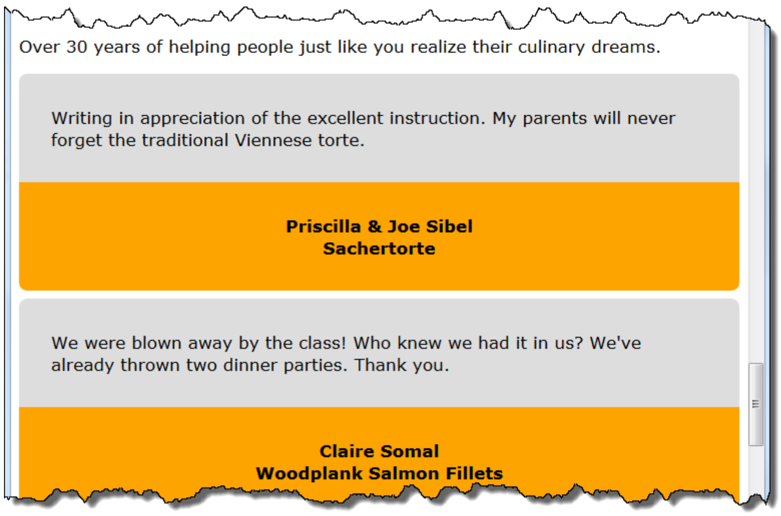
Over 30 years of helping people just like you realize their culinary dreams.

Writing in appreciation of the excellent instruction. My parents will never forget the traditional Viennese torte.

Priscilla & Joe Sibel
Sachertorte

We were blown away by the class! Who knew we had it in us? We've already thrown two dinner parties. Thank you.

23. Style the last paragraph inside each of the #testimonial <div>'s to have a background color of #fda400. Round the bottom corners to .5rem and remove any roundness on the top corners. Make the text bold and center-align it.



Over 30 years of helping people just like you realize their culinary dreams.

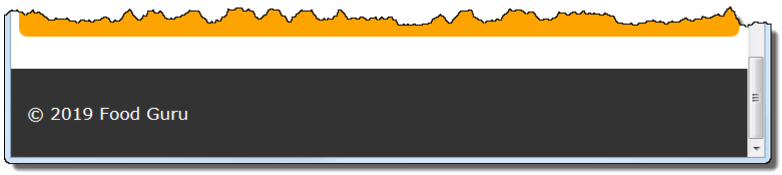
Writing in appreciation of the excellent instruction. My parents will never forget the traditional Viennese torte.

Priscilla & Joe Sibel
Sachertorte

We were blown away by the class! Who knew we had it in us? We've already thrown two dinner parties. Thank you.

Claire Somal
Woodplank Salmon Fillets

24. Style the footer to have a background color of #333, text color of white, a top margin of 1rem, and padding on all sides of 1rem.



© 2019 Food Guru

25. At this point, you should have 20 CSS rules and about 123 lines of code not counting the final blank line. Your page should match *Assets/Part 3 /Comps/02-basic-styling.jpg*.

Responsive nested grid layout

1. Create a second CSS file named **layout3.css**. Store it in your *css* subfolder and link to it in your HTML.
2. Replicate the full screen layout shown in *Assets/Part 3/Comps/ 03-full-screen.jpg*. This layout works down to about 870px wide.
    a. Every grid has a gutter (a gap) of 10px.
    b. The grid should have a maximum width of 1100px and should be centered in the viewport.
    c. Both the page header and footer are full bleed.
3. You may use numbered lines or grid areas.
4. Locate breakpoints and write appropriate media queries so that the layout is responsive. I found breakpoints at 870px and 550px worked for me. Yours may differ. Check the *Assets/Part 3/Comps/Breakpoints* folder for possible layouts at smaller sizes. *Note that improvements could be made to the layout if we used responsive images. But those are not part of this project.*
5. You must use a *support query* to ensure only browsers that fully support CSS Grid use the grid layout. *(Hint: CSS: Advanced Layouts with Grid course on Lynda.com, Chapter 4, Check For Grid Support)*
6. Validate all code.
7. Ensure all code is developer-friendly and adheres to our Code Formatting Guidelines.
8. Ensure the page displays and behaves as expected in all required browsers.

Submission

1. Zip your **last-first-grids-3** folder and submit via Moodle.