Google Search

* Link in upper-right to go to Image Search
* Link in upper-right to go to Image Search
* Should have a form and a button. “
  + Search bar details
    - This form should send info to Google’s actual search page.
    - The search bar should have rounded corners.
  + Button details:
    - Button default is google search.
    - Search button should be centred
    - Search button should be beneath the search bar

Google Image Search

* Link in upper-right to go to Google Search
* Link in upper-right to go to Advance Search

Google Advanced Search

* Link in upper-right to go to Google Search
* Link in upper right to go to Advance Search

**Display Attribute CSS**

* Specifies the display behaviour – the type of rendering box – of an element
* Default value taken from HTMl specification or browser style sheet. Default value in XML is inline.
* Values in attributes are:
  + Inline means on the same line as elements on the same level in DOM
  + Block means element starts on a new line and takes up the whole width
  + Contents means the container will disappear making the child elements children of the element the next level up in the DOM
  + Flex means display the element as a block-level flex container

**What is Flexbox?**

* Aims to efficiently lay out, align and distribute space among items in a container even when their sizes are unknown/dynamic
* Aims to do this by giving container the ability to alter its items width/height and order to best fill the available space. This is done to accommodate more display devices and screen sizes. Flex container will expand items to fill available free space or shrink them to prevent overflow.

1. How Flexbox works

* Flexbox is a module rather than a property. Involves a whole set of other properties some of which are meant to be set on the container. Container means parent element. A parent element with display flex is called a flex container. The children of that container would be called flex items.
* Terminology
  + Length called main-size
  + Main axis means axis across the length
  + Height called cross size.
  + Cross axis means axis across height
  + Main start means the start of the container from the left main end means the end of the container on the right
  + Cross start means the top of the container and cross end means the bottom of the container
  + Items are laid out along the main axis from main start to main end OR along the cross axis from cross start to cross end
  + Note: Main-axis and cross-axis not necessarily horizontal and vertical. Instead, it depends on the flex-direction property. Main axis will be the direction of the flex-direction property while cross axis will be perpendicular to the main-axis.
  + Main size would be flex item’s width or height whichever is the main direction is the item’s main size.

1. Properties for the parent (flex container)

* Display which is used to instantiate the flex container. Inline or block depends on the given value. Enables flex context for all direct children
* Flex direction establishes the main axis and defines the direction flex items are placed in a flex container. Flexbox is a single-direction layout concept. This means flex items are laid out either in horizontal or vertical columns no slants.
  + Options of flex-directions are:
    - Row (default) means directions are left to right
    - Row-reverse means items laid out right to left
    - Column which means same as row but top to bottom
    - Column-reverse which means bottom to top
* Flex-wrap which means allow items to wrap as needed. Flex-wrap: wrap
* Justify-content: Defines the alignment along the main axis. Helps distribute extra free space leftover when either all the flex items on a line are inflexible or are flexible but have reached max size. Exerts control over alignment of items when they overflow lines
  + Options are:
    - Flex-start: items are packed towards end of flex-direction
    - Flex-end: items are packed towards the start of the flex-direction
    - Start: items are packed towards the start of the writing-mode direction
    - End: items are packed towards the end of the writing-mode direction
    - Center: items are centred along the line
* Align-items: Defines the behaviour of how flex items are laid out along the cross axis on the current line. Basically the justify-content for cross axis.

Organization of Google Search Page

DOM layout

* Always remember to divide your DOM layout properly. This means figure out where to put divs. It is okay to nest divs within divs as long as the organization makes sense. However, really important to remember relationship between elements as a lot of html properties depend on these relationships
  + So plan the divs first
  + Remember properties sometimes within the same element depend on each other for example if a flex container has align-items: center but it does not have a height specified, no effect will take place as it does not know how to align along the cross-axis. Assuming that main-axis is horizontal and cross-axis is vertical.