=Als(IsLeeg([Fasering detail].[Projectcode]);"";Vervangen(Vervangen("<html>

<head>

  <script type=\"text/javascript\" src=\"[https://www.gstatic.com/charts/loader.js\](https://www.gstatic.com/charts/loader.js/)"></script>

  <script type=\"text/javascript\">

    google.charts.load('43', {'packages':['gantt']});

    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

      var data = new google.visualization.DataTable();

      data.addColumn('string', 'Fase');

      data.addColumn('string', 'Fase Name');

      data.addColumn('string', 'Resource');

      data.addColumn('date', 'Start Datum');

      data.addColumn('date', 'EindDatum');

      data.addColumn('number', 'Duur');

      data.addColumn('number', 'Percentage Gereed');

      data.addColumn('string', 'Afhankelijkheid');

      data.addRows**(["+[Max Concat Planning detail]+"]);**

      var options = {

        height: 600, width: 1000,

        gantt: {

          trackHeight: 30

        }

      };

      var chart = new google.visualization.Gantt(document.getElementById('chart\_div'));

      chart.draw(data, options);

    }

  </script>

</head>

<body>

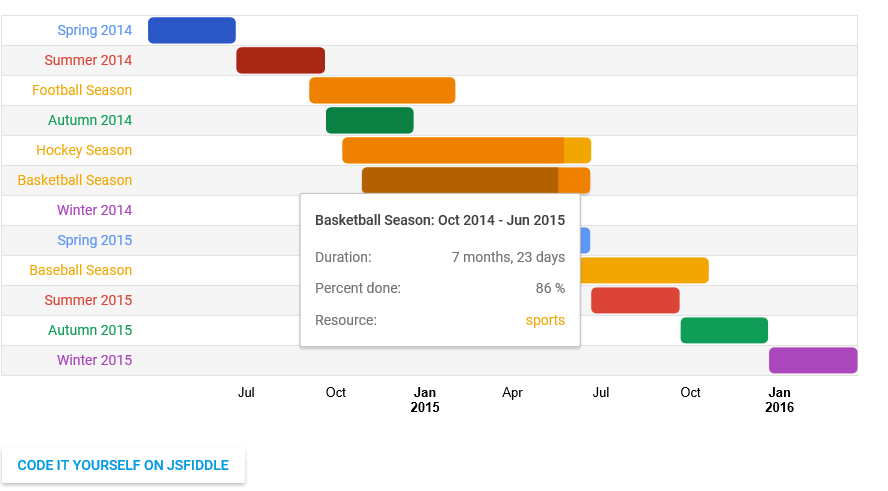
  <div id=\"chart\_div\"></div>

</body>

</html>";",['','','',new Date(), new Date(), null, 100, ''],]";"]");"''";"null"))

No dependencies

To create a Gantt chart that has no dependencies, make sure that the last value for each row in your DataTable is set to null.

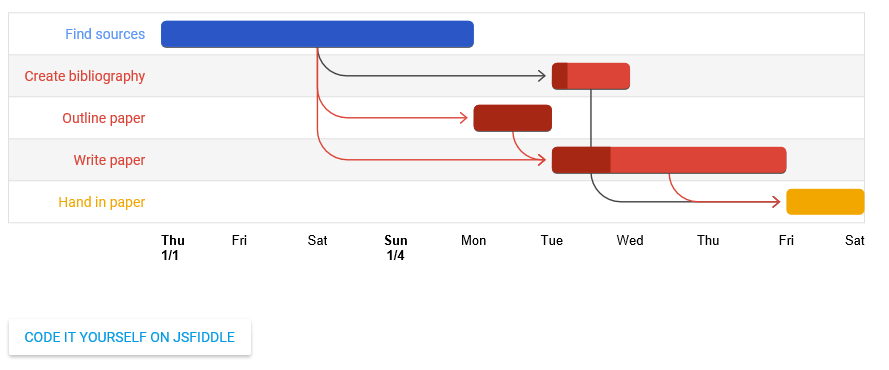


Onderkant formulier

<html>  
<head>  
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>  
  <script type="text/javascript">  
    google.charts.load('current', {'packages':['gantt']});  
    google.charts.setOnLoadCallback(drawChart);  
  
    function drawChart() {  
  
      var data = new google.visualization.DataTable();  
      data.addColumn('string', 'Task ID');  
      data.addColumn('string', 'Task Name');  
      data.addColumn('string', 'Resource');  
      data.addColumn('date', 'Start Date');  
      data.addColumn('date', 'End Date');  
      data.addColumn('number', 'Duration');  
      data.addColumn('number', 'Percent Complete');  
      data.addColumn('string', 'Dependencies');  
  
      data.addRows([  
        ['2014Spring', 'Spring 2014', 'spring',  
         new Date(2014, 2, 22), new Date(2014, 5, 20), null, 100, null],  
        ['2014Summer', 'Summer 2014', 'summer',  
         new Date(2014, 5, 21), new Date(2014, 8, 20), null, 100, null],  
        ['2014Autumn', 'Autumn 2014', 'autumn',  
         new Date(2014, 8, 21), new Date(2014, 11, 20), null, 100, null],  
        ['2014Winter', 'Winter 2014', 'winter',  
         new Date(2014, 11, 21), new Date(2015, 2, 21), null, 100, null],  
        ['2015Spring', 'Spring 2015', 'spring',  
         new Date(2015, 2, 22), new Date(2015, 5, 20), null, 50, null],  
        ['2015Summer', 'Summer 2015', 'summer',  
         new Date(2015, 5, 21), new Date(2015, 8, 20), null, 0, null],  
        ['2015Autumn', 'Autumn 2015', 'autumn',  
         new Date(2015, 8, 21), new Date(2015, 11, 20), null, 0, null],  
        ['2015Winter', 'Winter 2015', 'winter',  
         new Date(2015, 11, 21), new Date(2016, 2, 21), null, 0, null],  
        ['Football', 'Football Season', 'sports',  
         new Date(2014, 8, 4), new Date(2015, 1, 1), null, 100, null],  
        ['Baseball', 'Baseball Season', 'sports',  
         new Date(2015, 2, 31), new Date(2015, 9, 20), null, 14, null],  
        ['Basketball', 'Basketball Season', 'sports',  
         new Date(2014, 9, 28), new Date(2015, 5, 20), null, 86, null],  
        ['Hockey', 'Hockey Season', 'sports',  
         new Date(2014, 9, 8), new Date(2015, 5, 21), null, 89, null]  
      ]);  
  
      var options = {  
        height: 400,  
        gantt: {  
          trackHeight: 30  
        }  
      };  
  
      var chart = new google.visualization.Gantt(document.getElementById('chart\_div'));  
  
      chart.draw(data, options);  
    }  
  </script>  
</head>  
<body>  
  <div id="chart\_div"></div>  
</body>  
</html>

## Grouping resources

Tasks that are similar in nature can be grouped together using resources. Add a column of type string to your data (after the Task ID and Task Name columns), and make sure any tasks that should be grouped into a resource have the same resource ID. Resources will be grouped by color.

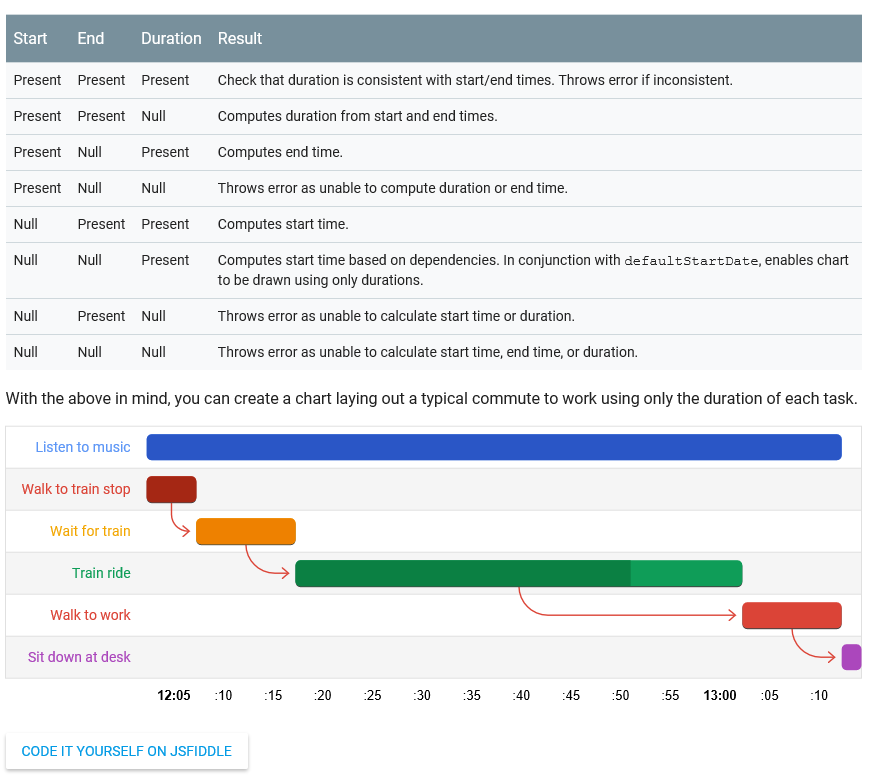


<html>  
<head>  
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>  
  <script type="text/javascript">  
    google.charts.load('current', {'packages':['gantt']});  
    google.charts.setOnLoadCallback(drawChart);  
  
    function daysToMilliseconds(days) {  
      return days \* 24 \* 60 \* 60 \* 1000;  
    }  
  
    function drawChart() {  
  
      var data = new google.visualization.DataTable();  
      data.addColumn('string', 'Task ID');  
      data.addColumn('string', 'Task Name');  
      data.addColumn('string', 'Resource');  
      data.addColumn('date', 'Start Date');  
      data.addColumn('date', 'End Date');  
      data.addColumn('number', 'Duration');  
      data.addColumn('number', 'Percent Complete');  
      data.addColumn('string', 'Dependencies');  
  
      data.addRows([  
        ['Research', 'Find sources', null,  
         new Date(2015, 0, 1), new Date(2015, 0, 5), null,  100,  null],  
        ['Write', 'Write paper', 'write',  
         null, new Date(2015, 0, 9), daysToMilliseconds(3), 25, 'Research,Outline'],  
        ['Cite', 'Create bibliography', 'write',  
         null, new Date(2015, 0, 7), daysToMilliseconds(1), 20, 'Research'],  
        ['Complete', 'Hand in paper', 'complete',  
         null, new Date(2015, 0, 10), daysToMilliseconds(1), 0, 'Cite,Write'],  
        ['Outline', 'Outline paper', 'write',  
         null, new Date(2015, 0, 6), daysToMilliseconds(1), 100, 'Research']  
      ]);  
  
      var options = {  
        height: 275  
      };  
  
      var chart = new google.visualization.Gantt(document.getElementById('chart\_div'));  
  
      chart.draw(data, options);  
    }  
  </script>  
</head>  
<body>  
  <div id="chart\_div"></div>  
</body>  
</html>

## Computing start/end/duration

Gantt charts accept three values relating to the duration of the task: a start date, an end date, and a duration (in milliseconds). If, for example, there is no start date, the chart can calculate the missing time based on the end date and the duration. The same goes for calculating the end date. If both the start and end date are given, the duration can be calculated between the two.

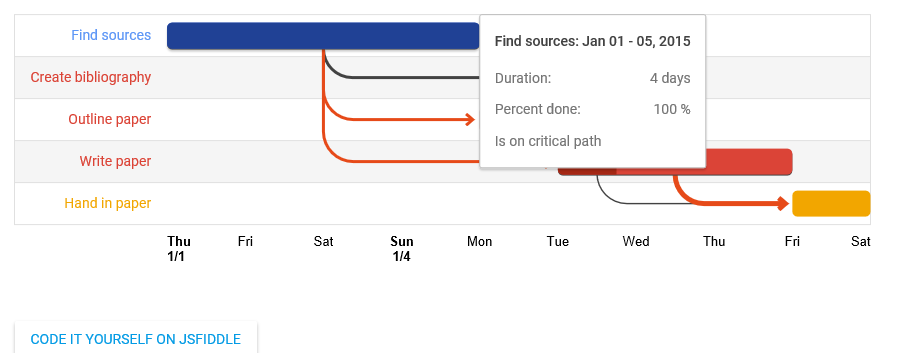
See the below table for a list of how Gantt handles the presence of start, end, and duration in different circumstances.



<html>  
<head>  
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>  
  <script type="text/javascript">  
    google.charts.load('current', {'packages':['gantt']});  
    google.charts.setOnLoadCallback(drawChart);  
  
    function toMilliseconds(minutes) {  
      return minutes \* 60 \* 1000;  
    }  
  
    function drawChart() {  
  
      var otherData = new google.visualization.DataTable();  
      otherData.addColumn('string', 'Task ID');  
      otherData.addColumn('string', 'Task Name');  
      otherData.addColumn('string', 'Resource');  
      otherData.addColumn('date', 'Start');  
      otherData.addColumn('date', 'End');  
      otherData.addColumn('number', 'Duration');  
      otherData.addColumn('number', 'Percent Complete');  
      otherData.addColumn('string', 'Dependencies');  
  
      otherData.addRows([  
        ['toTrain', 'Walk to train stop', 'walk', null, null, toMilliseconds(5), 100, null],  
        ['music', 'Listen to music', 'music', null, null, toMilliseconds(70), 100, null],  
        ['wait', 'Wait for train', 'wait', null, null, toMilliseconds(10), 100, 'toTrain'],  
        ['train', 'Train ride', 'train', null, null, toMilliseconds(45), 75, 'wait'],  
        ['toWork', 'Walk to work', 'walk', null, null, toMilliseconds(10), 0, 'train'],  
        ['work', 'Sit down at desk', null, null, null, toMilliseconds(2), 0, 'toWork'],  
  
      ]);  
  
      var options = {  
        height: 275,  
        gantt: {  
          defaultStartDateMillis: new Date(2015, 3, 28)  
        }  
      };  
  
      var chart = new google.visualization.Gantt(document.getElementById('chart\_div'));  
  
      chart.draw(otherData, options);  
    }  
  </script>  
</head>  
<body>  
  <div id="chart\_div"></div>  
</body>  
</html>

## Critical path

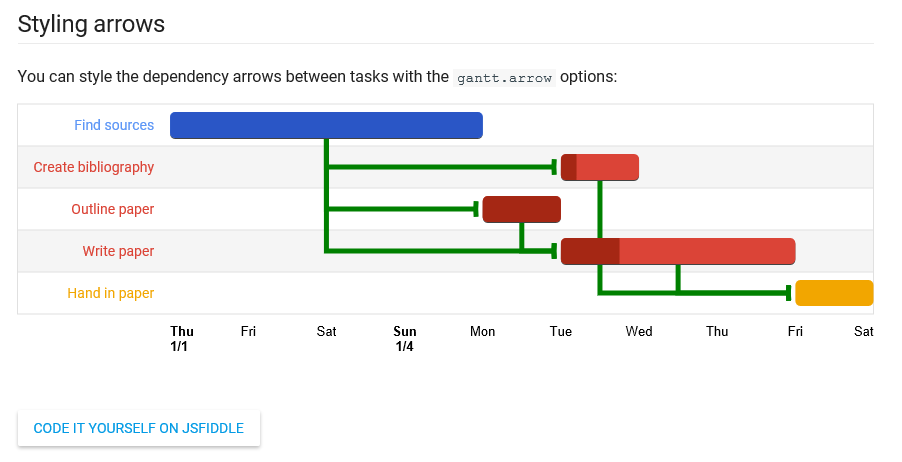
The critical path in a Gantt chart is the path, or paths, that directly affect the finish date. The critical path in Google Gantt charts is colored red by default, and can be customized using the criticalPathStyle options. You can also turn off the critical path by setting criticalPathEnabled to false.



      var options = {  
          height: 275,  
          gantt: {  
            criticalPathEnabled: true,  
            criticalPathStyle: {  
              stroke: '#e64a19',  
              strokeWidth: 5  
            }  
          }  
        };

## Styling arrows

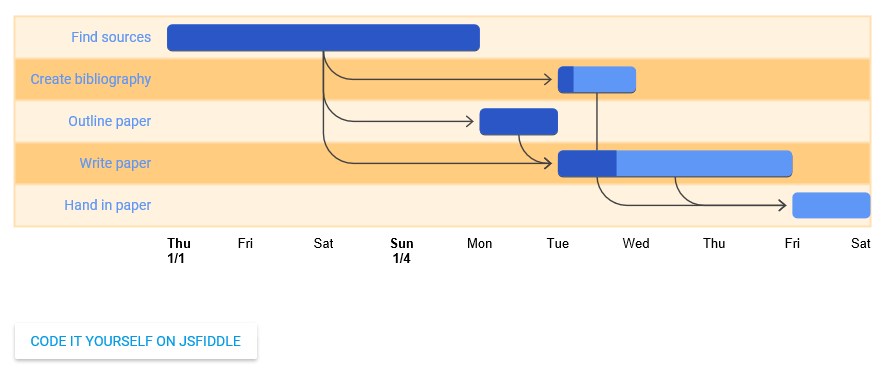
You can style the dependency arrows between tasks with the gantt.arrow options:



      var options = {  
          height: 275,  
          gantt: {  
            criticalPathEnabled: false, // Critical path arrows will be the same as other arrows.  
            arrow: {  
              angle: 100,  
              width: 5,  
              color: 'green',  
              radius: 0  
            }  
          }  
        };

## Styling tracks

Grid styling is handled by a combination of innerGridHorizLine, innerGridTrack, and innerGridDarkTrack. By setting only the innerGridTrack, the chart will calculate a darker color for the innerGridDarkTrack, but by setting only the innerGridDarkTrack, the innerGridTrack will use its default color and will not calculate a lighter color.



    var options = {  
        height: 275,  
        gantt: {  
          criticalPathEnabled: false,  
          innerGridHorizLine: {  
            stroke: '#ffe0b2',  
            strokeWidth: 2  
          },  
          innerGridTrack: {fill: '#fff3e0'},  
          innerGridDarkTrack: {fill: '#ffcc80'}  
        }  
      };

## Loading

The google.charts.load package name is "gantt".

  google.charts.load("current", {packages: ["gantt"]});

The visualization's class name is google.visualization.Gantt.

  var chart = new google.visualization.Gantt(container);

## Data format

**Rows:** Each row in the table represents a task.

**Columns:**

