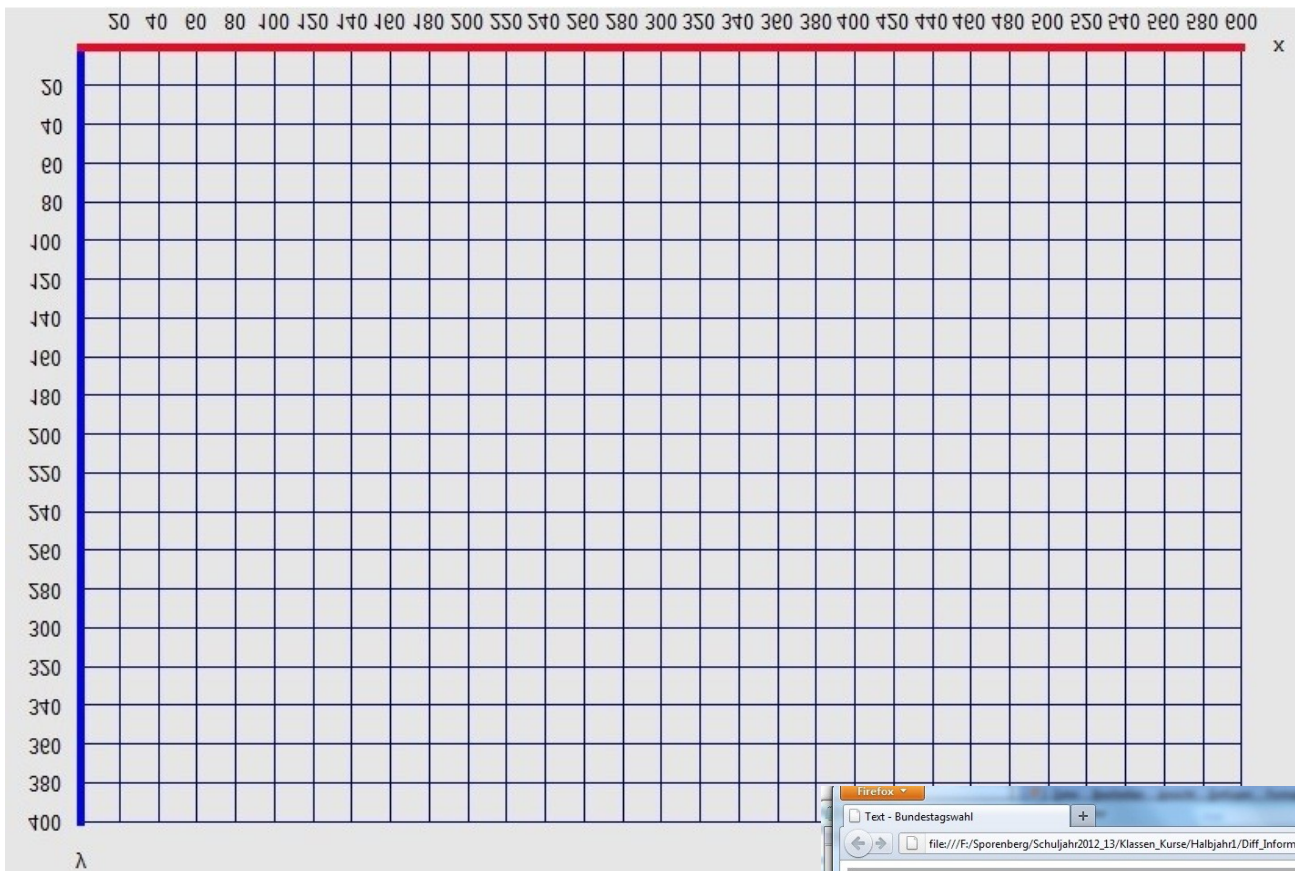
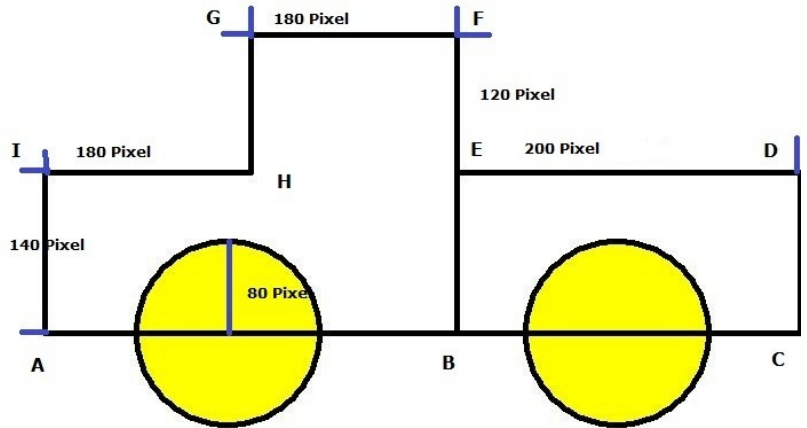


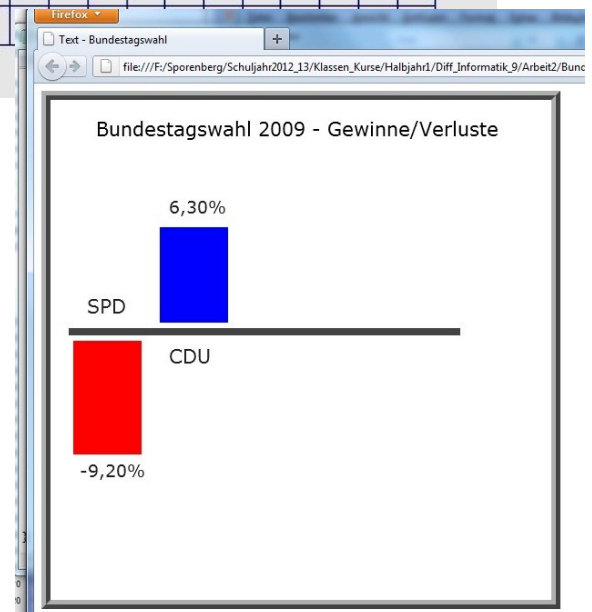
1.Aufgabe: Programmiere folgende Figur mit Hilfe von Javascript im Canvas-Element von HTML5.

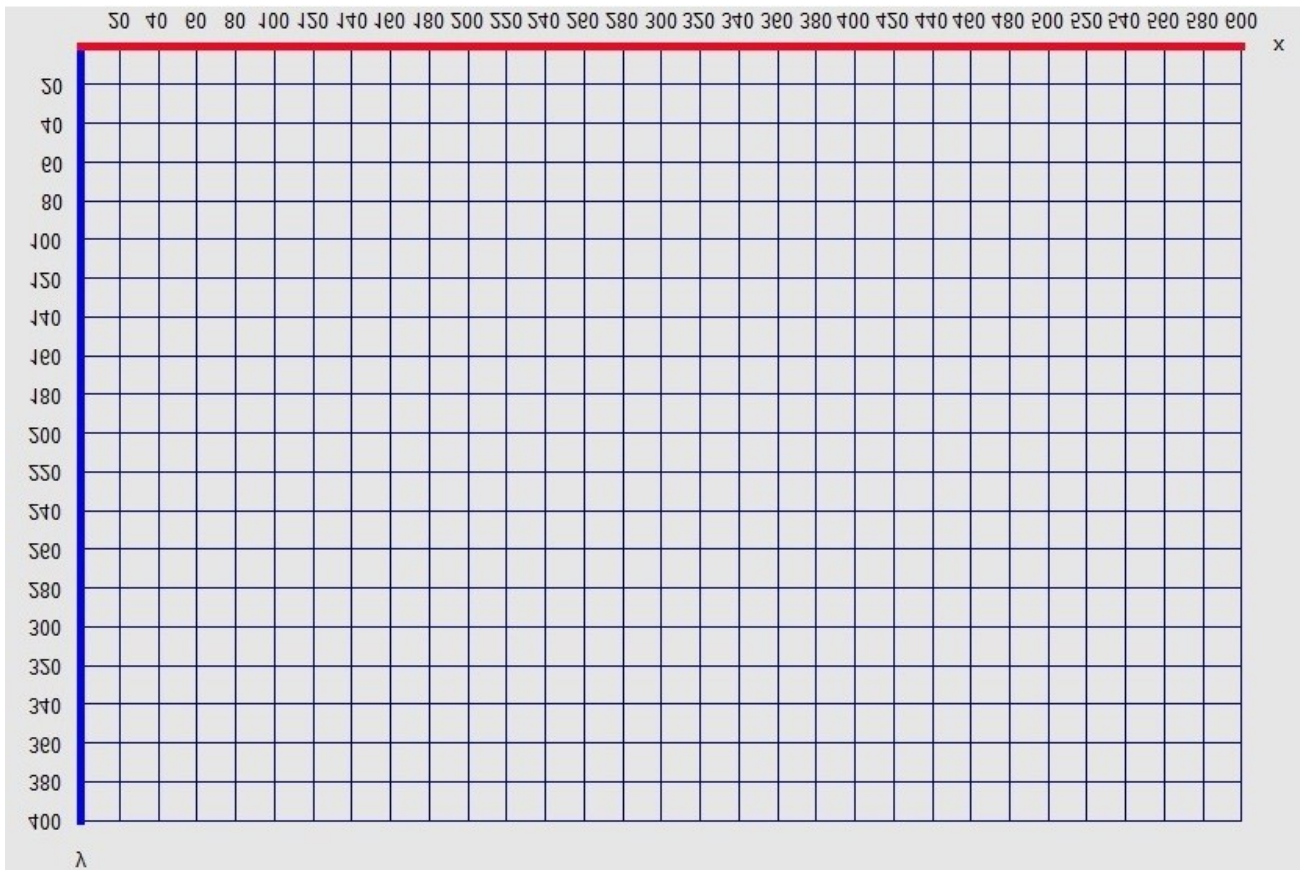
Übertrage die Umriss des LKW's entsprechend den Angaben von Länge und Breite in das Koordinatenkreuz und gib die Koordinaten der Eckpunkte (A, B, C, D, E, F, G, H und I) an. Schreibe dann ein Javascript-Programm, das die Figur auf der Zeichenfläche canvas darstellt.



2.Aufgabe: Für die Darstellung von Gewinn/Verlust bei der Bundestagswahl 2009 werden Säulen entsprechend der Prozentzahl und Zunahme/Abnahme mit einer Beschriftung der Parteien und der entsprechenden Prozentzahl gebildet. Schreibe dazu die entsprechenden Javascript-Zeilen, die die Graphik im canvas-Element ausgibt.

Bevor das Programm geschrieben wird, fertige für die Angaben der Koordinaten der Säulen eine Skizze im Koordinatensystem an. Dabei sollen die Höhen der Säulen entsprechend den Prozentzahlen dargestellt werden (1% entspricht 10 Pixel). Die Breite der Säulen beträgt 70 Pixel.





3. Aufgabe: Ein Kreis soll sich auf der canvas von oben nach unten bewegen. Schreibe dazu die entsprechende Animation in der Scriptsprache Javascript.

Beispiel: Hier wird ein Kreis von links nach rechts bewegt.

```
function draw(){
  var canvas = document.getElementById('canvas');
  if(canvas.getContext){
    var context = canvas.getContext('2d');
    animate(context, 50, 50); }}

function animate(context, ax, ay){
  setTimeout(function(){
    context.clearRect(0, 0, 400, 400);
    context.beginPath();
    context.fillStyle = "#EEE8AA";
    context.lineWidth = 10;
    context.strokeStyle = "#DAA520";
    context.arc(ax ,ay, 30, 0, 2 * Math.PI, true);
    context.stroke();
    context.fill();
    ax=ax+1;
    ay=50;
    if (ax>290) {ax=50};
    self.animate(context, ax, ay);}, 4);}
```

4.Aufgabe: Ein Rechteck der Breite 50 Pixel soll von oben nach unten wachsend bis zu einer Länge von 150 Pixel auf der canvas animiert werden. Schreibe dazu die entsprechende Animation in der Scriptsprache Javascript.

Im Beispiel wächst das Rechteck von unten nach oben:

```
function draw(){
  var canvas = document.getElementById('canvas');
  if(canvas.getContext){
    var context = canvas.getContext('2d');
    animate(context, 200, 200); } }

function animate(context, ay, Hoehey){
  setTimeout(function(){
    context.clearRect(0, 0, 800, 400);
    context.fillStyle = "rgb(0, 0, 255)";
    context.fillRect(50, ay, 50, Hoehey);
    context.lineWidth = "4";
    context.strokeStyle = "#cdcdcd";
    context.strokeRect(50, ay, 50, Hoehey);
    ay=ay-2;
    Hoehey=Hoehey+2;
    if(ay<100){return};
    self.animate(context, ay, Hoehey); }, 4);}
```

5.Aufgabe: Ein Kreis mit dem Radius 30 Pixel soll auf der rechteckigen canvas-Fläche von oben links diagonal nach unten rechts bewegt werden. Der Kreis soll also auf der Diagonalen des Canvas-Rechtecks bewegt werden. Schreibe dazu die entsprechende Animation in der Scriptsprache Javascript.

Befehle in canvas-HTML5

context.arc(x, y, Radius, startWinkel, endWinkel, Richtung);

Kreis mit dem **Radius**, Kreismittelpunkt (x,y), **startWinkel** und **endWinkel**. **Richtung** gibt an, im Uhrzeigersinn (=false) oder gegen den Uhrzeigersinn (=right).

Für die Textausgabe:

```
context.fillStyle = "blue";
context.font = "48pt Helvetica";
context.fillText("Reflection", xko, yko);
```

Alle anderen Befehle können aus den Programmbeispielen entnommen werden.

Hinweis: Die canvas-Fläche hat eine Breite von 800 Pixel und eine Höhe von 400 Pixel.

Lösung:

1. Aufgabe:

```
function draw(){
  var canvas = document.getElementById('canvas');
  var context = canvas.getContext('2d');
  if(canvas.getContext){

    context.fillStyle = "#ffff00";
    context.lineWidth=10;
    context.moveTo( 0+20, 260+20 );
    context.lineTo( 360+20,260+20);
    context.lineTo( 560+20,260+20);
    context.lineTo( 560+20,120+20);
    context.lineTo( 360+20,120+20);
    context.lineTo( 360+20,0+20);
    context.lineTo( 180+20,0+20);
    context.lineTo( 180+20,120+20);
    context.lineTo( 0+20,120+20);
    context.lineTo( 0+20,260+20);
    context.stroke();

    context.arc(180+20, 260+20, 80, 0, 2*Math.PI,true);
    context.stroke();

    context.arc(440+20, 260+20, 80, 0, 2*Math.PI,true);
    context.stroke();
  }}

```

Die Koordinaten der Punkte:

A(20/280), B(380/280), C(580/280), D(580/140), E(380/140), F(380/20), G(200/20),
H(200/140), I(20/140).

2. Aufgabe:

```
function draw(){
  var canvas = document.getElementById('testcanvas');
  var context = canvas.getContext('2d');
  if(canvas.getContext){

    context.beginPath();

    context.font = "16pt Verdana";
    context.fillStyle = "#000000";
    context.fillText("Bundestagswahl 2009 - Gewinne/Verluste", 50, 40);

    context.font = "16pt Verdana";
    context.fillStyle = "#222222"; // text color
    context.fillText("SPD", 40, 235);
    context.font = "14pt Verdana";
    context.fillText("-9,20%", 32, 415);

    context.lineWidth=8;
    context.strokeStyle = "#444444";
    context.moveTo( 20, 255 );
    context.lineTo(450, 255 );
  }}

```

```

context.stroke();

context.fillStyle = "#ff0000";
context.fillRect(25,265, 75, 125);
context.fill();

context.font = "16pt Verdana";
context.fillStyle = "#222222"; // text color
context.fillText("CDU", 130, 290);
context.font = "14pt Verdana";
context.fillText("6,30%", 130, 125);

context.fillStyle = "#0000ff";
context.fillRect(120,140, 75, 105);
context.fill();

context.closePath();
  }}

```

3. Aufgabe:

Das Programm muss nur bei den Variablen verändert werden. Die x-Koordinate ax muss konstant bleiben, dafür muss die y-Koordinate vergrößert werden, z.B. ay=ay+1
Das Programm kann demnach fast identisch übernommen werden. Die Änderungen sind rot dargestellt:

```

function draw(){
  var canvas = document.getElementById('canvas');
  if(canvas.getContext){
    var context = canvas.getContext('2d');
    animate(context, 50, 50); }}

function animate(context, ax, ay){
  setTimeout(function(){
    context.clearRect(0, 0, 400, 400);
    context.beginPath();
    context.fillStyle = "#EEE8AA";
    context.lineWidth = 10;
    context.strokeStyle = "#DAA520";
    context.arc(ax ,ay, 30, 0, 2 * Math.PI, true);
    context.stroke();
    context.fill();
    ax=50;
    ay=ay+3;
    if (ay>300) {ay=300};
    self.animate(context, ax, ay);}, 10);}

```

4. Aufgabe:

Die Lösung ergibt sich aus der vorliegenden Programmtext. Hier muss lediglich die Länge des Rechtecks vergrößert werden. Die x- und y-Koordinate bleiben gleich.

Das Programm kann demnach fast identisch übernommen werden. Die Änderungen sind rot dargestellt:

```

function draw(){
  var canvas = document.getElementById('canvas');
  if(canvas.getContext){
    var context = canvas.getContext('2d');
    animate(context, 50, 0);  } }

  function animate(context, ay, Hoehey){
    setTimeout(function(){
      context.clearRect(0, 0, 800, 400);
      context.fillStyle = "rgb(0, 0, 255)";
      context.fillRect(50, ay, 50, Hoehey);
      context.lineWidth = "4";
      context.strokeStyle = "#cdcdcd";
      context.strokeRect(50, ay, 50, Hoehey);
      Hoehey=Hoehey+2;
      if(Hoehey>150){Hoehey=150};
      self.animate(context, ay, Hoehey); }, 4);}

```

5.Aufgabe:

Auch hier kann der Programmcode, der zur Aufgabe 3 gehört, fast identisch übernommen werden. Die Änderungen beziehen sich auf die x- und y-Koordinate des Kreismittelpunktes. Hier müssen beide Koordinaten erhöht werden.

Die Änderungen sind rot dargestellt:

```

function draw(){
  var canvas = document.getElementById('canvas');
  if(canvas.getContext){
    var context = canvas.getContext('2d');
    animate(context, 50, 50);  } }

function animate(context, ax, ay){
  setTimeout(function(){
    context.clearRect(0, 0, 400, 400);
    context.beginPath();
    context.fillStyle = "#EEE8AA";
    context.lineWidth = 10;
    context.strokeStyle = "#DAA520";
    context.arc(ax ,ay, 30, 0, 2 * Math.PI, true);
    context.stroke();
    context.fill();
    ax=ax+1;
    ay=ay+1;
    if (ax>300) {ax=300;ay=300};
    self.animate(context, ax, ay);}, 4);}

```