

```
Blockly.Language.webgl_a09_html = {
  category: 'WebGL',
  helpUrl: 'http://www.example.com/',
  init: function() {
    this.setColour(330);
    this.appendValueInput("HTML")
      .setAlign(Blockly.ALIGN_CENTRE)
      .appendTitle("HTML");
    this.appendStatementInput("HTML");
    this.setPreviousStatement(true);
    this.setNextStatement(true);
    this.setTooltip("");
    this.appendValueInput("VALUE")
      .appendTitle(new Blockly.FieldImage("../media/html.svg", 64, 48));
  }
};

Blockly.JavaScript.webgl_a09_html = function() {
  var value_html = Blockly.JavaScript.valueToCode(this, 'HTML',
Blockly.JavaScript.ORDER_ATOMIC);
  var statements_html = Blockly.JavaScript.statementToCode(this, 'HTML');
  var code = '&lt;html&gt;/' * + value_html + '&lt; \html&gt;' * /
  return code;
};
```



```

Blockly.Language.webgl_a09_body = {
  category: 'WebGL',
  helpUrl: 'file:///localhost/C:/blockly/demos/webgl/webgl_a09_headerhtml5.help.html',
  init: function () {
    this.setColour(65);
    this.appendStatementInput("BODY")
      .appendTitle("Body");
    this.appendStatementInput("NAME");
    this.appendStatementInput("NAME");
      this.appendDummyInput()
        .appendTitle("canvas id=")
        .appendTitle(new Blockly.FieldTextInput("canv"), "canv")
    this.setPreviousStatement(true);
    this.setNextStatement(true);
    this.setTooltip("");
  }
};

Blockly.JavaScript.webgl_a09_body = function() {
  var statements_body = Blockly.JavaScript.statementToCode(this, 'BODY');
  var statements_name = Blockly.JavaScript.statementToCode(this, 'NAME');
  var statements_name = Blockly.JavaScript.statementToCode(this, 'NAME');
  var code = '&lt;body&gt;&lt;' + statements_body + '&gt;&lt;/body&gt;\n';
  return code;
};

```



```
Blockly.Language.webgl_a09_header = {
  category: 'WebGL',
  helpUrl:
'file:///localhost/C:/blockly/demos/webgl/webgl_a09_headerhtml5.help.html',
  init: function() {
    this.setColour(65);
    this.appendStatementInput("HEADER")
      .appendTitle("Header");
    this.setOutput(true);
    this.setTooltip("");
  }
};
Blockly.JavaScript.webgl_a09_header = function() {
  var statements_header = Blockly.JavaScript.statementToCode(this,
'HEADER');
  // TODO: Assemble JavaScript into code variable.
  var code = '\n<header>\n' + value_header + '\n</header>';
  // TODO: Change ORDER_NONE to the correct strength.
  return [code, Blockly.JavaScript.ORDER_NONE];
};
```



```

Blockly.Language.webgl_a09_title = {
  category: 'WebGL',
  helpUrl: 'file://localhost/C:/blockly/demos/webgl/webgl_a09_headerhtml5.help.html',
  init: function() {
    this.setColour(160);
    this.appendDummyInput()
      .appendTitle("title")
      .appendTitle(new Blockly.FieldTextInput("Blockly"), "NAME");
    this.setPreviousStatement(true);
    this.setNextStatement(true);
    //this.appendValueInput("TITLE")
    // .appendTitle("Title");
    // this.setOutput(true);
    this.setTooltip("");
  }
};

Blockly.JavaScript.webgl_a09_title = function() {
  var value_title = Blockly.JavaScript.valueToCode(this, 'TITLE', Blockly.JavaScript.ORDER_ATOMIC);
  // TODO: Assemble JavaScript into code variable.
  //var code = '\n<title>\n;' + statements_title + '\n</title>';
  var code = "<title>" + value_title.toString() + "< Blockly ></canvas>";
  //return [code, Blockly.JavaScript.ORDER_NONE];
  return code;
};

```



```

Blockly.Language.a09_img = {
  category: 'WebGL',
  helpUrl:
'file:///localhost/C:/blockly/demos/webgl/webgl_a09_headerhtml5.help.html',
  init: function() {
    this.setColour(280);
    this.appendDummyInput()
      .appendTitle("img")
      .appendTitle(new Blockly.FieldTextInput("canvas"), "NAME");
    this.setPreviousStatement(true);
    this.setNextStatement(true);
    this.setTooltip(""); }
};

Blockly.JavaScript.a09_canvas = function() {
  var text_name_canvas = this.getTitleValue('NAME');
var code = "<canvas id=" + text_name_canvas.toString() + " width=\"120\"
height=\"120\" ></canvas>";
  canvas_name = text_name_canvas;
  return code;
};

```



```

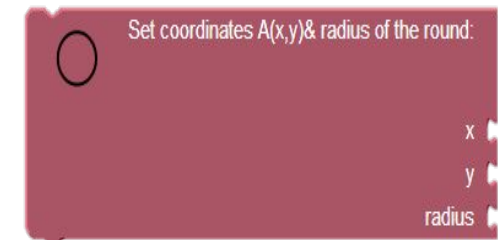
Blockly.Language.webgl_a09_circle = {
  category: 'WebGL_2d',
  helpUrl: '../webgl/webgl_a08_headerhtml5.help.html',
  init: function() {
    this.setColour(350);
    this.appendDummyInput()
      .appendTitle(new
Blockly.FieldImage("../media/html1.svg", 64, 48))
      .appendTitle("Set coordinates A(x,y)& radius of the
round:");
    this.appendValueInput("x")
      .setCheck(Number)
      .setAlign(Blockly.ALIGN_RIGHT)
      .appendTitle("x");
    this.appendValueInput("y")
      .setCheck(Number)
      .setAlign(Blockly.ALIGN_RIGHT)
      .appendTitle("y");
    this.appendValueInput("r")
      .setCheck(Number)
      .setAlign(Blockly.ALIGN_RIGHT)
      .appendTitle("radius");
    this.setPreviousStatement(true);
    this.setNextStatement(true);
    this.setTooltip("");
  }
};

```

```

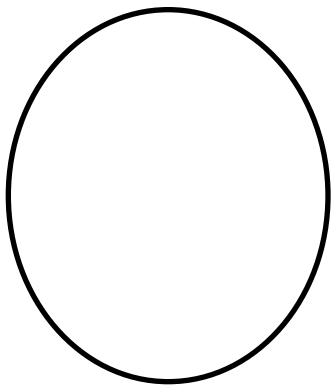
Blockly.JavaScript.webgl_a09_circle = function() {
  var value_canv_id =
Blockly.JavaScript.valueToCode(this, 'canv_id',
Blockly.JavaScript.ORDER_ATOMIC);
  var value_x = Blockly.JavaScript.valueToCode(this, 'x',
Blockly.JavaScript.ORDER_ATOMIC);
  var value_y = Blockly.JavaScript.valueToCode(this, 'y',
Blockly.JavaScript.ORDER_ATOMIC);
  var value_w = Blockly.JavaScript.valueToCode(this,
'radius', Blockly.JavaScript.ORDER_ATOMIC);
  // TODO: Assemble JavaScript into code variable.
  var code="var ctx = document.getElementById("+
value_canv_id + ").getContext('2d');\n"
code += 'ctx.save();\n';
code += 'ctx.beginPath();\n';
code += 'ctx.translate('+ value_x+', '+ value_y+');\n';
code += 'ctx.scale('+value_w +', 1);\n';
code += "\n gl.arc(" + parseInt(text_x) + ",
"+parseInt(text_y)+", "+parseInt(text_radius)+", 0, 2 *
Math.PI, false);";
code += 'ctx.restore();\n';
code += 'ctx.closePath();\n';
code += 'ctx.fill();\n';
return code;
};

```



# JavaScript код программы и ее результат

```
html>
<head>
<title> Blockly</title>
</head>
<body>
  <canvas id='canvas' width="1000" height="800">
</canvas>
  <script>
if (window.WebGLRenderingContext){
var c = document.getElementById('canvas');
  var gl = c.getContext('2d');
  gl.beginPath();
  gl.arc(75,75, 37, 0, 2 * Math.PI, false);
  gl.lineWidth = 1;
  gl.strokeStyle = '#FFFFFFF';
  gl.stroke();
}
</script>
</body>
```



# Блок-представление программы

