Improving tooling for novice programmers

By Harrison Shoebridge (u6960999).
Supervised by Ben Swift.
learning to program is hard
let fish = null;

function setup() {
    createCanvas(windowWidth, windowHeight)
    fish = {x: 2, y: 3; color = "green"; size = 30; }
}

function draw() {
    background("blue")
    fill(fish.color)
    circle(fish.x, fish.y, fish.size)
}
let x = 0;

function setup() {
  createCanvas(windowWidth, windowHeight);
}

function draw() {
  rect(x, 0, 100, 100);
  x += "1";
}
let x = 0;

function setup() {
  createCanvas(windowWidth, windowHeight);
  noStroke();
}

function draw() {
  background("white");

  fill("blue");
  circle(100, 100, 100);
  fill("green");
  rect(100, 100, 100, 100);
}
tutors
a good scalable tutoring experience
computer scientists like building things
what's out there?

- iSnap
- different methods of generating hints
- software for Java that enhances error messages
they’re good, but don’t help me too much with my specific problem.

- focus on Java or Python -- very different languages than JS
- often leave out a big class of errors: “domain” errors
the path forward?
collect data
allows us to make empirical decisions

- know what errors students learning p5.js and JS come across
- which ones they spend their time on
- begin the process of finding out how to resolve these in a way which mimics a high quality tutor
demo
important considerations

● how much data do we collect? (on key events: errors and compilation)
● how can we not burden users?
● how much do we ask of them?
what we can do with this data?

- know exactly what students at struggling with week to week -- and address it
- improve documentation of p5.js to help beginners
- create a model for classifying future errors