New Data Types - Vectors

- Arithmetic operations work component-wise
- Built-in functions operate on vectors component-wise
  - For example: `cos`, `sin`, `abs`, `sqrt`, ...
  - Also available `length`, `distance`, `normalize`, `dot`, `cross`, ...

```
vec2 a = vec2(13, 37);
vec2 b = vec2(85, 19);
vec2(a.x * b.x, a.y * b.y) == a * b;
```

```
vec2 a = vec2(13, 37);
vec3 b = vec3(85, 19, 08);
vec2(a.x * b.x, a.y * b.y) == a * b.xy
```

New Data Types - Matrices

- {ε d}mat{2 3 4}{ε x2 x3 x4}
- Examples:
  - mat2 (= mat2x2): float, 2 columns, 2 rows
  - dmat3 (= dmat3x3): double, 3 columns, 3 rows
  - mat3x4: float, 3 columns, 4 rows
- Matrices (on default) are **column-major** (but can be changed)
- Accessors:

```
mat3x4 matrix;
vec3 col1 = matrix[0];  // First column
float val1 = matrix[2][1]; // Third column, second row
float val2 = matrix[2].x; // val2 == val1
```

- Arithmetic operations behave as expected
  - matαxβ ∗ vecβ   = vecβ
  - matαxβ ∗ matαxβ = matαxβ
  - vecα ∗ matβxδ   compile error