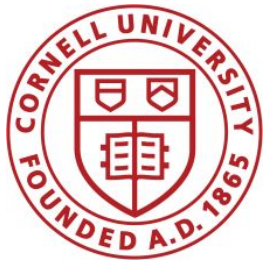


DSFA

Spring 2018

Lecture 13

Control

Announcements

Recipes

INGREDIENTS

2 1/4 cups all-purpose flour

1 teaspoon baking soda

1 teaspoon salt

1 cup (2 sticks) butter, softened

3/4 cup granulated sugar

3/4 cup packed brown sugar

1 teaspoon vanilla extract

2 large eggs

2 cups (12-oz. pkg.) NESTLÉ® TOLL HOUSE® Semi-Sweet Chocolate Morsels

1 cup chopped nuts

IN THIS RECIPE

INSTRUCTIONS

VIEW: TEXT | VIDEO

PREHEAT oven to 375° F.

COMBINE flour, baking soda and salt in small bowl. Beat butter, granulated sugar, brown sugar and vanilla extract in large mixer bowl until creamy. Add eggs, one at a time, beating well after each addition. Gradually beat in flour mixture. Stir in morsels and nuts. Drop by rounded tablespoon onto ungreased baking sheets.

BAKE for 9 to 11 minutes or until golden brown. Cool on baking sheets for 2 minutes; remove to wire racks to cool completely.

PAN COOKIE VARIATION: Preheat oven to 350° F. Grease 15 x 10-inch jelly-roll pan. Prepare dough as above. Spread into prepared pan. Bake for 20 to 25 minutes or until golden brown. Cool in pan on wire rack. Makes 4 dozen bars.

SLICE AND BAKE COOKIE VARIATION:
PREPARE dough as above. Divide in half; wrap in waxed paper. Refrigerate for 1 hour or until firm. Shape each half into 15-inch log; wrap in wax paper. Refrigerate for 30 minutes.* Preheat oven to 375° F. Cut into 1/2-inch-thick slices; place on ungreased baking sheets. Bake for 8 to 10 minutes or until golden brown. Cool on baking sheets for 2 minutes; remove to wire racks to cool completely. Makes about 5 dozen cookies.

Recipe instructions

COMBINE flour, baking soda and salt in small bowl. Beat butter, granulated sugar, brown sugar and vanilla extract in large mixer bowl until creamy. Add eggs, one at a time, beating well after each addition. Gradually beat in flour mixture. Stir in morsels and nuts. Drop by rounded tablespoon onto ungreased baking sheets.

BAKE for 9 to 11 minutes or until golden brown. Cool on baking sheets for 2 minutes; remove to wire racks to cool completely.

Algorithm

Rules or *a recipe* for performing computation

Ideas we see in cookie recipe:

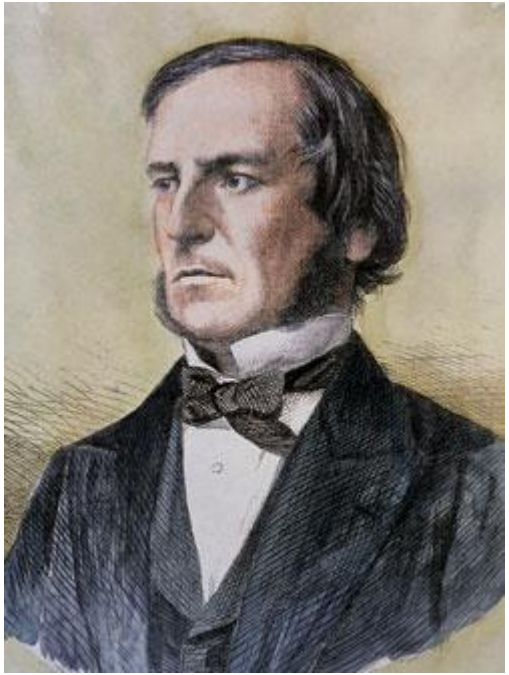
- Iteration: do something many times
- Conditionals: decide whether something is true, and maybe do something different
- Variability or randomness: some tasks might not be completely predictable

Comparison

George Boole

The Laws of Thought (1854)

No general method for the solution of questions in the theory of probabilities can be established which does not explicitly recognise, not only the special numerical bases of the science, but also those **universal laws of thought which are the basis of all reasoning**, and which, whatever they may be as to their essence, are at least mathematical as to their form.



Comparison Operators

The result of a comparison expression is a **`bool`** value

```
x = 2                    y = 3
```

Assignment statements

```
x > 1                    x > y                    y >= 3

x == y                   x != 2                   2 < x < 5
```

Comparison expressions

(Demo)

Combining Comparisons

Boolean operators can be applied to **`bool`** values

```
a = True        b = False
```

Evaluate to True

```
not b           a or b              a and not b
```

```
a and b         not (a or b)        b and b
```

Evaluate to False

(Demo)

Aggregating Comparisons

Summing an array or list of bool values will count the True values only.

```
1     + 0      + 1              == 2
True + False + True            == 2
sum([1    , 0     , 1    ))   == 2
sum([True, False, True))   == 2
```

(Demo)

Random Selection

Random Selection

`np.random.choice`

- Selects at random
- with replacement
- from an array
- a specified number of times

```
np.random.choice(some_array, sample_size)
```

(Demo)

Discussion Question

```
d6 = np.arange(1,6+1)
```

What results from evaluating the following 2 expressions? Are they the same? Do they describe the same process?

```
np.random.choice(d6, 1000) + np.random.choice(d6, 1000)
```

```
2 * np.random.choice(d6, 1000)
```

Control Statements

Control Statements

These statements *control* the sequence of computations that are performed in a program

- The keywords `if` and `for` begin control statements
- The purpose of `if` is to define computations that can choose different behaviors
- The purpose of `for` is to perform a computation for every element in a collection

(Demo)