CMSC201
Computer Science I for Majors

Lecture 23 –
Hexadecimal and Color Printing

Last Class We Covered

- ASCII values
- Short circuit evaluation

- Project 3

Any Questions from Last Time?

Today's Objectives

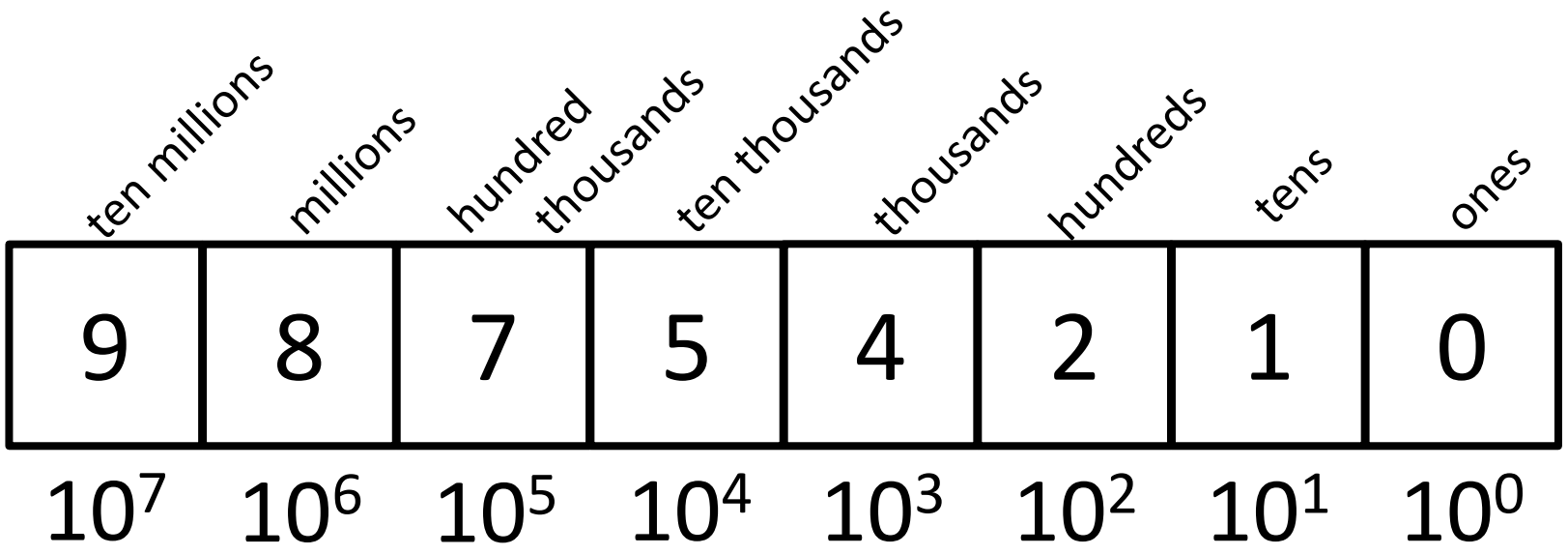
- To understand more about how data is represented inside the computer
  - Hexadecimal numbers
- To show how to print in color

- To learn a possibly useful method for your Project 3

Hexadecimal Numbers

Decimal Representation

- Decimal uses 10 digits
  - Decimal, *deci* = 10
  - The digits used are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9

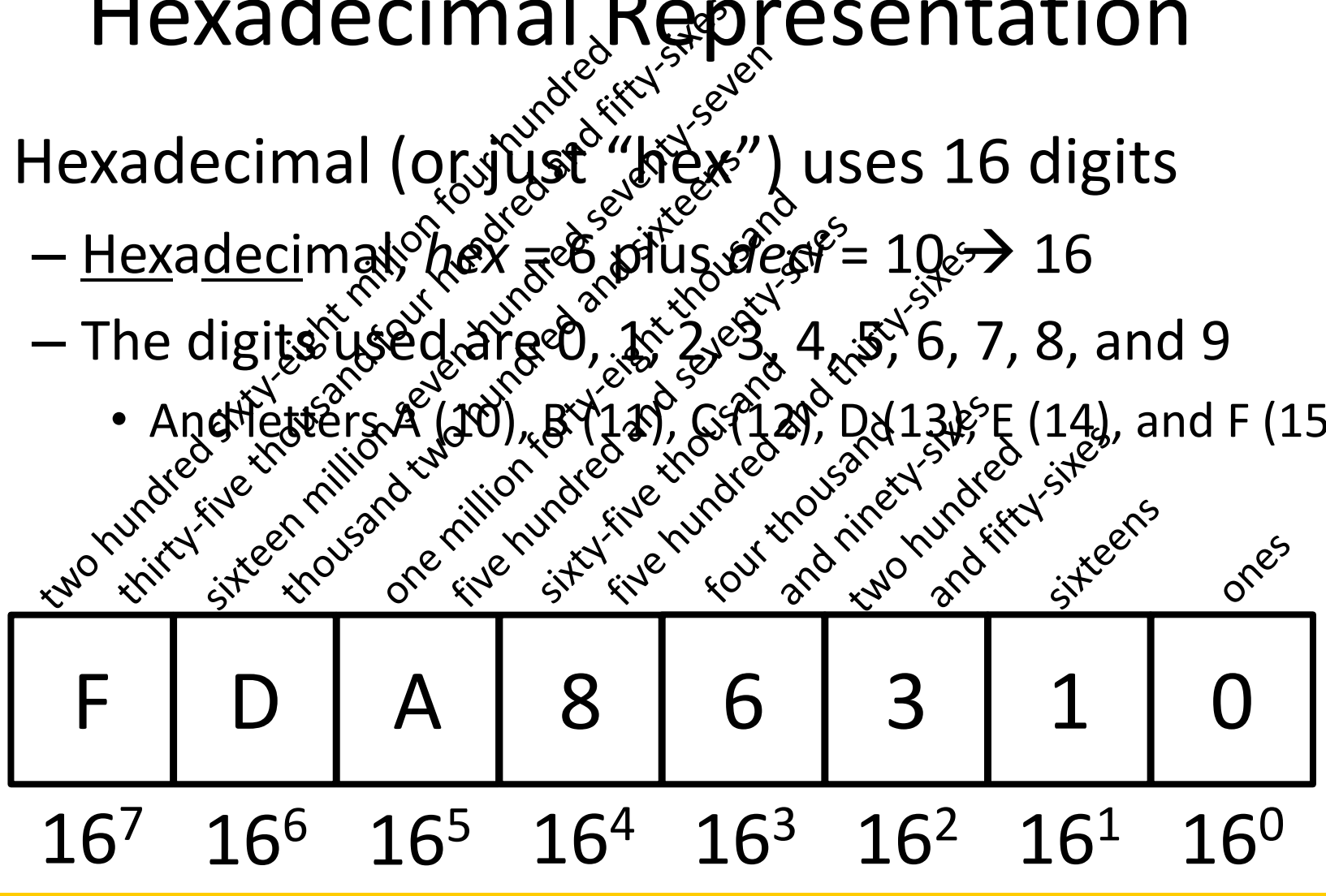


Binary Representation

- Binary uses 2 digits
  - Binary, *bi* = 2
  - The digits used are 0 and 1

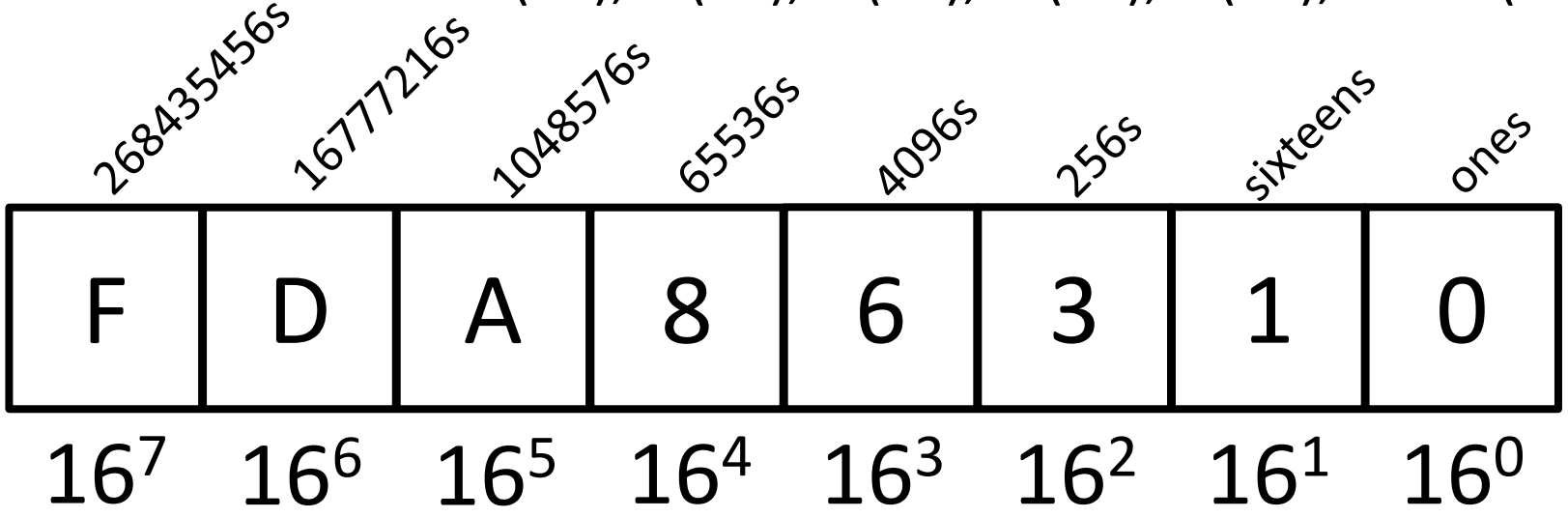
Hexadecimal Representation

- Hexadecimal (or just "hex") uses 16 digits
  - Hexadecimal, *hex* = 6 plus *dec* = 10 → 16
  - The digits used are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9
    - And letters A (10), B (11), C (12), D (13), E (14), and F (15)



Hexadecimal Representation

- Hexadecimal (or just "hex") uses 16 digits
  - Hexadecimal, *hex* = 6 plus *deci* = 10 → 16
  - The digits used are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9
    - And letters A (10), B (11), C (12), D (13), E (14), and F (15)



Hex to Binary Conversion

- A hexadecimal digit can be easily represented as four digits of binary (with leading zeros)

Hex	Binary	Hex	Binary	Hex	Binary	Hex	Binary
0	0000	4	0100	8	1000	C	1100
1	0001	5	0101	9	1001	D	1101
2	0010	6	0110	A	1010	E	1110
3	0011	7	0111	B	1011	F	1111

- This makes conversion very simple
  - **7A0F** becomes **0111 1010 0000 1111**
  - **1100 0010 0110 1001** becomes **C269**

Hex to Decimal Conversion

- Possible to convert between decimal and hex
  - But it requires calculating out multiples of 16
- Simpler to make a "side trip" to binary as an in-between step when converting
  - 240 becomes **`1111 0000`** becomes **`F0`**
    - **`F0`** is equal to $(15 * 16^1) + (0 * 16^0) = 240 + 0 = 240$
  - **`7D`** becomes **`0111 1101`** becomes 125
    - **`7D`** is equal to $(7 * 16^1) + (13 * 16^0) = 112 + 13 = 125$

Number System Notation

- Because number systems share a subset of the same digits, it may be confusing which is which
  - For example, what is the value of 10?
    - In decimal it's 10, in binary it's 2, and in hex it's 16

- To prevent this, numbers may often be prefixed with **`0b`**, **`0d`**, or **`0x`** (binary, decimal, hex):
  - **`0b1100`** is binary, and has a value of 12
  - **`0x15`** is hexadecimal, and has a value of 21

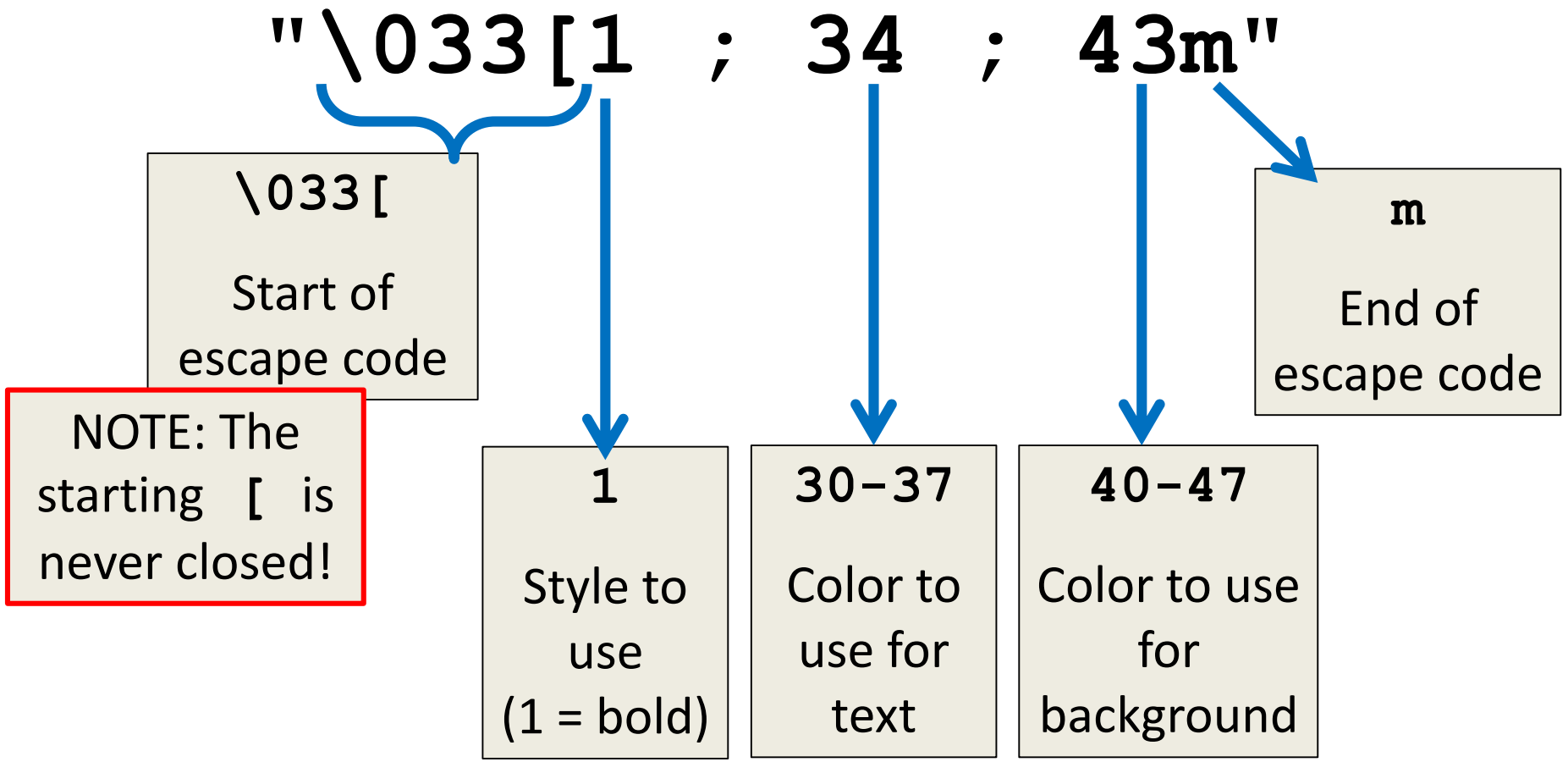
Printing in Color

ANSI Escape Codes

- To change the color of the background and text, we can use ANSI escape codes
  - Works in many languages, not just Python

- To use the codes, simply use **`print()`**
  - Just like "**`\t`**" turns into a tab, these won't be "printed," but will change how the text looks
  - For example, `print("\033[1;34;43m")` changes text to blue, and background to yellow

Syntax of ANSI Escape Color Codes



Color Values and Reset

- The colors available are black, red, green, yellow, blue, magenta, cyan, and white
  - For text color, they are 30 – 37, in order
  - For background, they are 40 – 47, in order
- This is a perfect use for a dictionary!
  - Store the color name as the key, and the number as the value; no need to memorize the numbers
- To reset to default colors, use `"\033[0m"`

Example Usages

```
START  = "\033[1;"
RESET  = "\033[0m"
COLORS = {'black': '30', 'red': '31', 'green': '32', 'yellow': '33',
          'blue': '34', 'magenta': '35', 'cyan': '36', 'white': '37'}
>>> print("\033[1;36;40m" + "Dogs are great, even in cyan" + RESET)
Dogs are great, even in cyan
>>> print(START + COLORS["red"] + ";44m" + "Red on blue!" + RESET)
Red on blue!
>>> print("\033[1;30;42m")
>>> print("Until it's reset, it prints black on green from now on!")
Until it's reset, it prints black on green from now on!
>>> print("\033[0m")

>>> print("\033[1;32;45m" + "Why would you do this?" + RESET)
Why would you do this?
```

Function to Print In Color

- Printing in color can be very useful when trying to distinguish different types of output
  - Like debugging vs normal program output

- We've provided a function for you under the "Livecoding" on the course website
  - Feel free to use it in your Project 3 for debugging
  - (Do NOT make your output hard to read, though!)
    - (Your TA will take off points if it's obnoxious!)

Possibly Helpful Method

The `.isdigit()` Method

- Works on a string, returns **`True`** or **`False`**

```
>>> numDogs = "101"
>>> numDogs.isdigit()
   True
>>> "3.14".isdigit()
   False
>>> "7".isdigit()
   True
>>> "201 ".isdigit()
   False
```

Daily CS History

- Hemachandra
  - Was a Jain scholar, poet, and polymath
  - Lived from 1088 to 1173 in India
  - Came up with the Fibonacci sequence 50 years before Fibonacci
    - While coming up with different long and short syllable combinations for traditional poetry

  - https://youtu.be/_32rgS8ClKw?t=1m54s

Announcements

- Project 3 design is due on Friday, May 4th
  - Project itself is due on Friday, May 11th
- Survey #3 out on Monday, May 7th
  - Final exam metacognition quiz out on BB same day
- Course evaluations are out now
  - Please complete them
- Final exam is Friday, May 18th from 6 to 8 PM

Final Exam Locations

- Find your room ahead of time!
- **Engineering 027** - Sections 8, 9, 10, 11, 12
  Section 6
- **Meyerhoff 030** - Sections 2, 3, 4, 5
  Sections 14, 15, 16, 17, 30

Image Sources

- Hemachandra:
  - https://commons.wikimedia.org/wiki/File:Hemachandra.gif