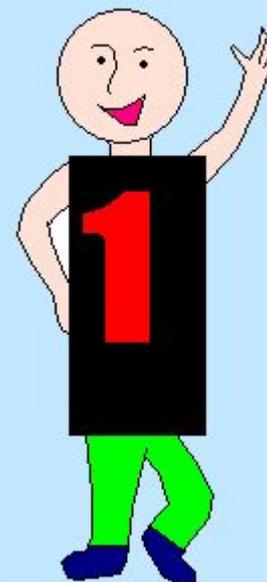
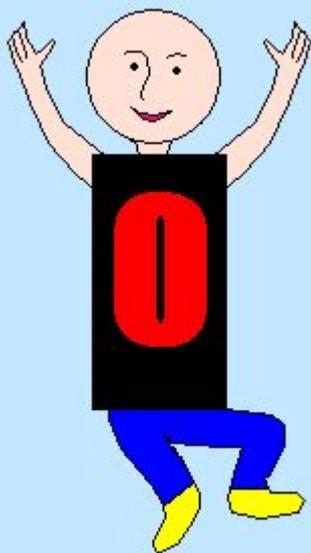




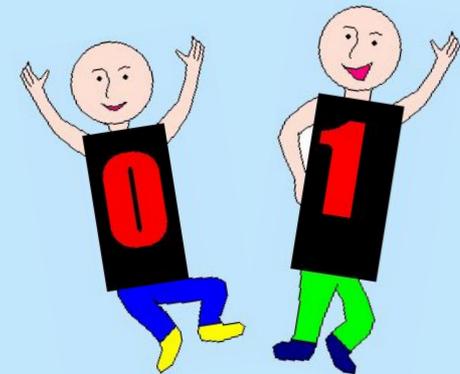
# Цифровые данные





# Хранение информации в компьютере

- Машинную память удобно представить в виде листа в клетку.
- В каждой «клетке» хранится только одно из двух значений: ноль или единица.
- Каждая «клетка» памяти называется битом.
- Цифры 0 и 1, хранящиеся в «клетках» памяти компьютера, называются значениями битов.



1	0	0	1	1	0	0	0
1	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0
1	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
1	0	1	0	0	1	0	0
0	1	0	0	0	0	1	1
1	0	0	1	1	0	0	1
1	0	1	0	0	0	0	0
0	0	0	1	0	1	0	1
1	1	0	1	0	0	1	0



# Двоичное кодирование

- Числовая информация
- Текстовая информация
- Графическая информация





# Десятичная позиционная система счисления

**Десятичная** – потому что десять единиц одного разряда составляют одну единицу старшего разряда; для записи чисел используются десять цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

**Позиционная** – потому, что одна и та же цифра получает разные количественные значения в зависимости от позиции, которую она занимает в записи числа.





# Немного математики

Рассмотрим числовой ряд:

**1, 10, 100, 1 000, 10 000, 100 000, ...**

Любое целое число можно представить в виде суммы разрядных слагаемых – единиц, десятков, сотен, тысяч и т.д., записанных в этом ряду:

$$1652 = 1 \times 1\ 000 + 6 \times 100 + 5 \times 10 + 2 \times 1$$

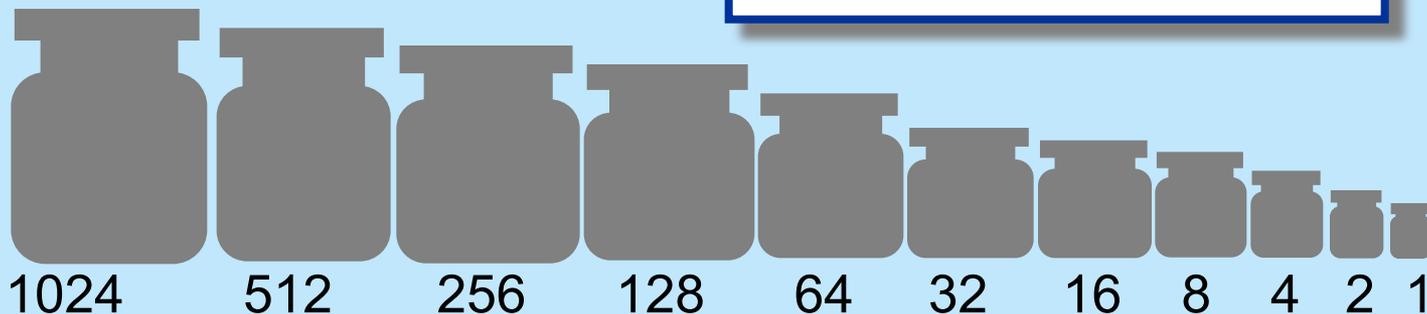
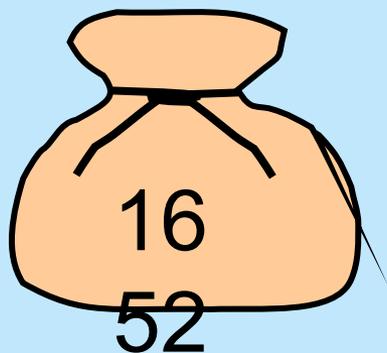
А теперь рассмотрим другой ряд:

**1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, ...**



# Поиграем в магазин

В нашем распоряжении есть чашечные весы и 10 разных гирек. Попробуем с их помощью уравновесить груз весом 1652 г.

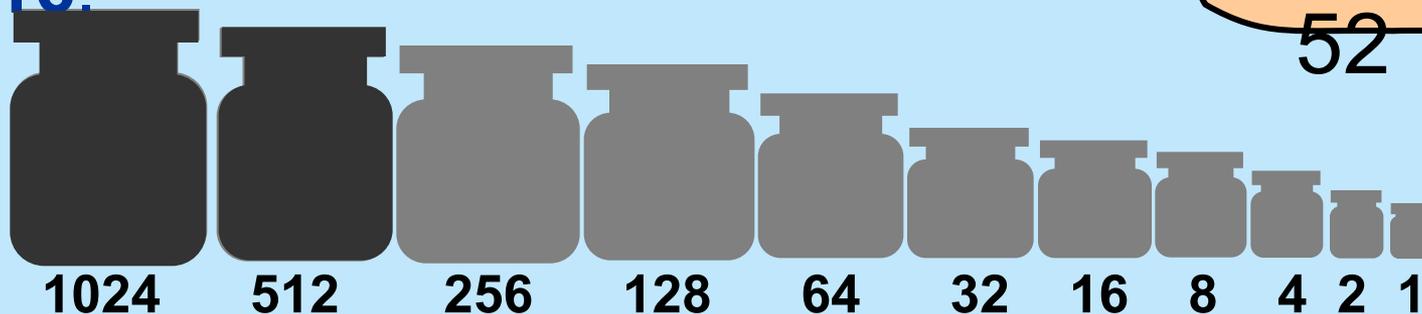
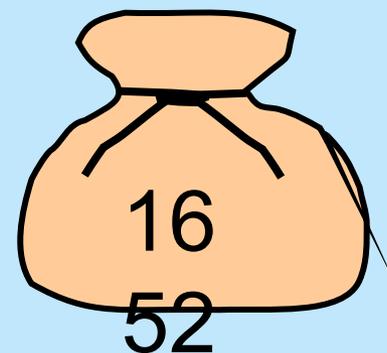




# Метод разностей

На одну чашу весов ставим груз, а на другую – гирьку с весом, ближайшим к весу груза, но не превышающим его. Найдем разность:  
 $1652 - 1024 = 628$ .

Найдем гирьку с весом, ближайшим к полученной разности, но не превышающим ее:  $628 - 512 = 116$ .





# Метод разностей

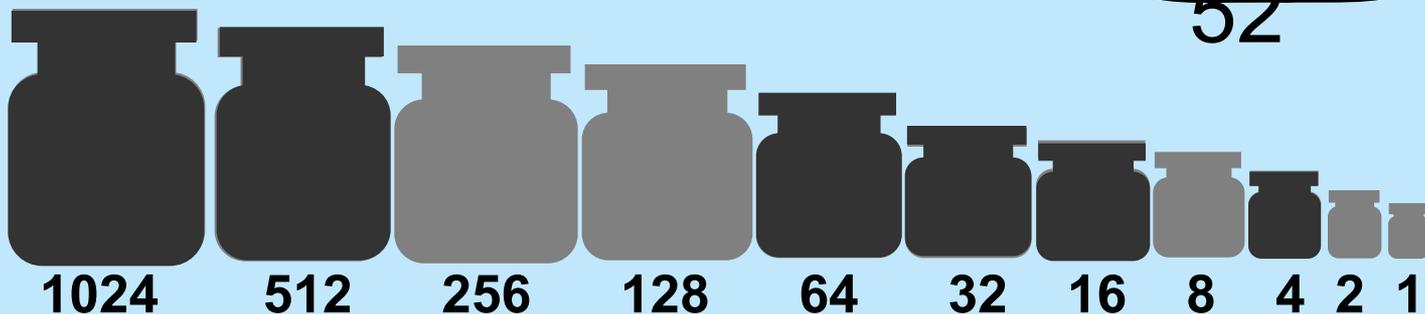
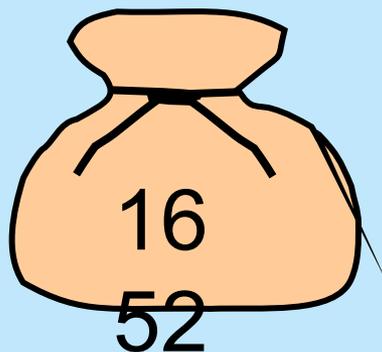
$$1652 - 1024 = 628$$

$$628 - 512 = 116$$

$$116 - 64 = 52$$

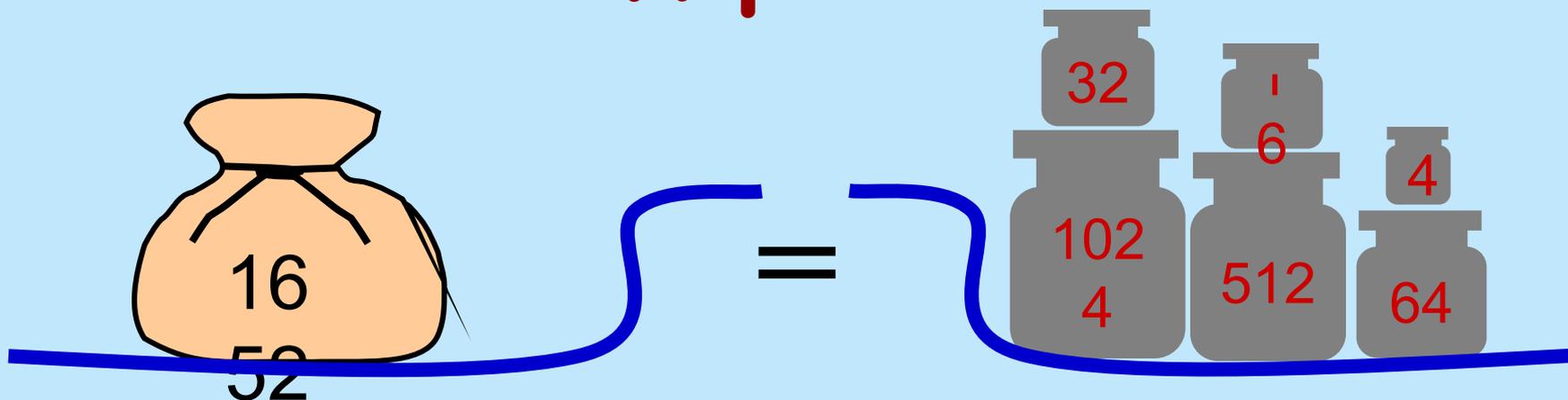
$$52 - 32 = 20$$

$$20 - 16 = 4$$





# Метод разностей



$$1652 = 1024 + 512 + 64 + 32 + 16 + 4 = 1 \times 1024 + 1 \times 512 +$$

$$+ 0 \times 256 + 0 \times 128 + 1 \times 64 + 1 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2$$

$$+ + 0 \times 1$$

$$1652 \rightarrow 11001110100$$





# Двоичная система счисления

$$1652 = 1024 + 512 + 64 + 32 + 16 + 4 = 1 \times 1024 + 1 \times 512 + \\ + 0 \times 256 + 0 \times 128 + 1 \times 64 + 1 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 \\ + + 0 \times 1$$

$$1652_{10} = 11001110100_2$$

Мы представили число в двоичной позиционной системе счисления:

**двоичной** – потому что две единицы одного разряда составляют одну единицу старшего разряда; для записи чисел используются две цифры: 0 и 1;

**позиционной** – потому, что одна и та же цифра получает разные количественные значения в зависимости от позиции, которую она занимает в записи числа.



# Перевод целых десятичных чисел в двоичную систему

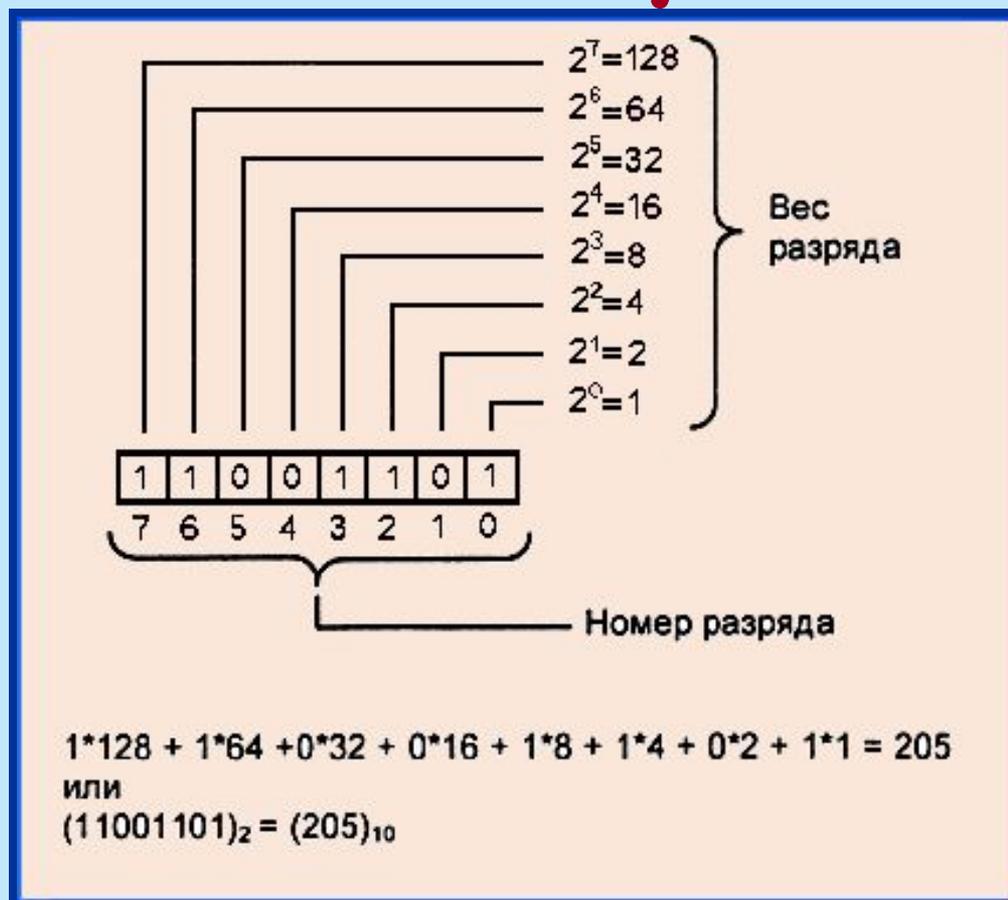
- Разделить целое десятичное число на 2. Остаток записать.
- Если полученное частное не меньше 2, то продолжать деление.
- Двоичный код десятичного числа получается при последовательной записи последнего частного и всех остатков, начиная с последнего.

23 | 2  
22 | 11 | 2  
1 | 10 | 5 | 2  
1 | 1 | 4 | 2 | 2  
1 | 2 | 1  
0

$23_{10} = 10111_2$



# Перевод целых десятичных чисел в двоичную систему





# Историческая справка



Лейбниц Готфрид Вильгельм (1646 - 1716), немецкий ученый, заложивший основы двоичной системы счисления





# Двоичное кодирование текстовой информации

Мы знаем, как перевести целое десятичное число в двоичный код.

А если каждому символу текста присвоить номер и по известным правилам перевести это номер в двоичный код?

**T → 210 → 11010010**

Именно эта идея положена в основу двоичного кодирования текстовой информации!





# Сколько нужно символов?

В текстах мы используем:

- прописные и строчные русские буквы **Аа Бб Вв ...**
- прописные и строчные латинские буквы **Аа Вb Сс ...**
- знаки препинания **! , ? . ...**
- цифры **1 2 3 ...**
- знаки арифметических операций **+ - × ...**
- другие символы **( [ \ ...**

**Достаточно 256 различных символов.**



# Кодовые таблицы

Соответствие символов и кодов задается с помощью специальных кодовых таблиц.

В кодовых таблицах каждому **символу** ставится в соответствие уникальная **цепочка из восьми нулей и единиц**.

Символ	Десятичный код	Двоичный код
!	33	00100001
...	...	...
А	192	11000000
Б	193	11000001
В	194	11000010

Арабская (Windows)

Балтийская (ISO)

Балтийская (Windows)

Центральноевропейская (DOS)

Центральноевропейская (ISO)

Центральноевропейская (Windows)

Китайская упрощенная (GB2312)

Китайская упрощенная (HZ)

Китайская традиционная (Big5)

Кириллица (DOS)

Кириллица (ISO)

Кириллица (KOI8-R)

Кириллица (KOI8-U)

Греческая (ISO)

Греческая (Windows)

Иврит (DOS)



# Кодовая таблица в системе Windows

Символ	Десятичный код	Двоичный код	Символ	Десятичный код	Двоичный код
Пробел	32	00100000	0	48	00110000
!	33	00100001	1	49	00110001
*	42	00101010	2	50	00110010
+	43	00101011	3	51	00110011
,	44	00101100	4	52	00110100
-	45	00101101	5	53	00110101
.	46	00101110	6	54	00110110
/	47	001011110	7	55	00110111
=	61	00111101	8	56	00111000
?	63	00111111	9	57	00111001
А	192	11000000	Р	208	11010000
Б	193	11000001	С	209	11010001
В	194	11000010	Т	210	11010010
Г	195	11000011	У	211	11010011
Д	196	11000100	Ф	212	11010100
Е	197	11000101	Х	213	11010101
Ж	198	11000110	Ц	214	11010110
З	199	11000111	Ч	215	11010111
И	200	11001000	Ш	216	11011000
Й	201	11001001	Щ	217	11011001
К	202	11001010	Ъ	218	11011010
Л	203	11001011	Ы	219	11011011
М	204	11001100	Ь	220	11011100
Н	205	11001101	Э	221	11011101
О	206	11001110	Ю	222	11011110
П	207	11001111	Я	223	11011111



# Двоичное кодирование графической информации

Графическое изображение можно разбить

- 1) крошечные фрагменты;
- 2) простейшие геометрические объекты.

На этом основано два варианта двоичного кодирования графической информации:

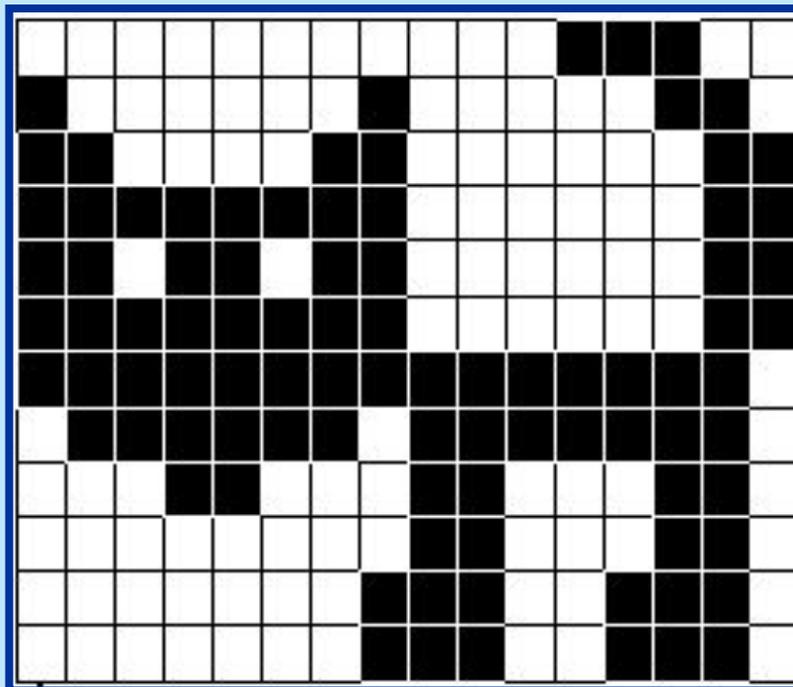
- растровый;
- векторный.





# Черно-белое изображение

```
0000000000011100
1000000100000110
1100001100000011
1111111100000011
1101101100000011
1111111100000011
1111111111111110
0111111011111110
0001100011000110
0000000011000110
0000000111001110
0000000111001110
```



0 – белая клетка

1 – черная клетка



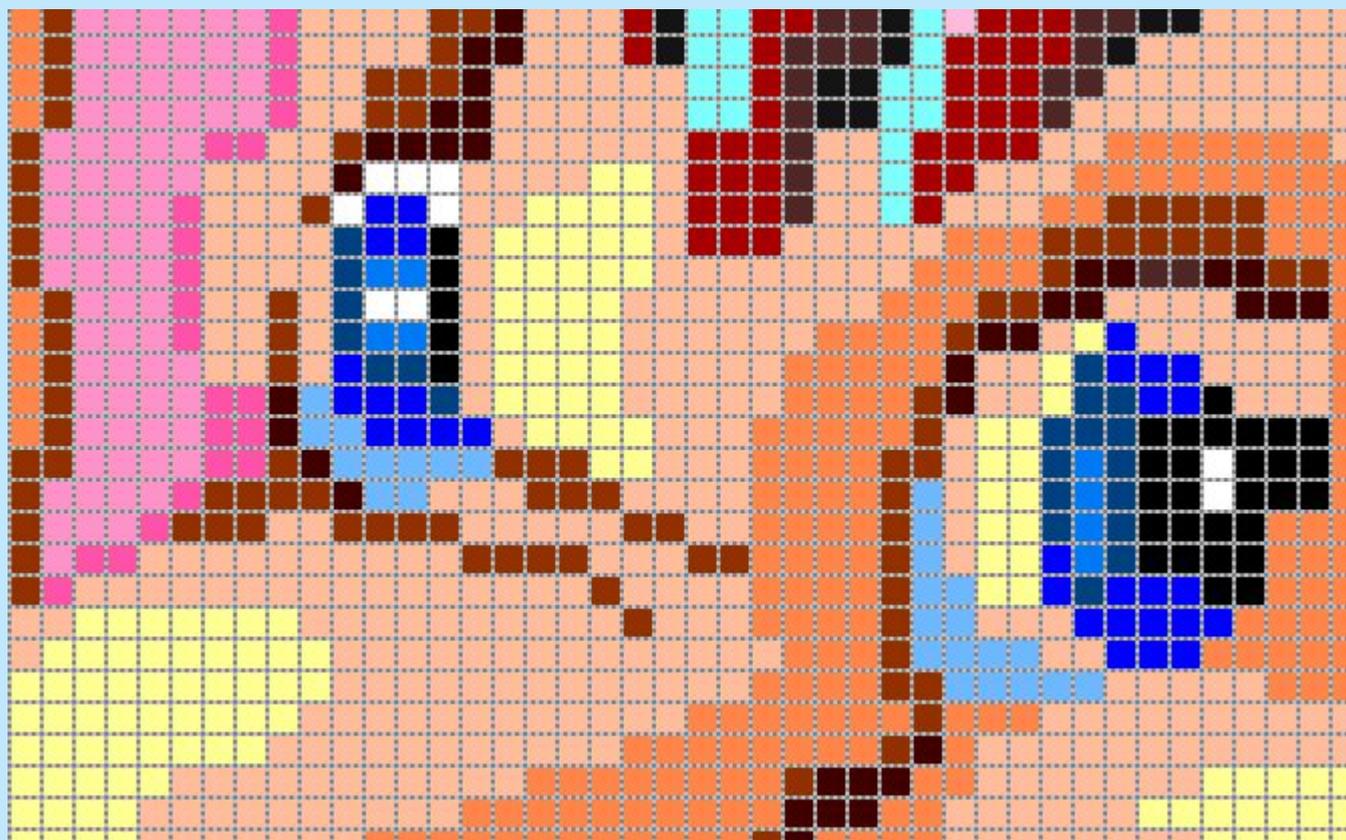
# Цветное изображение





# Цветное изображение

Каждый пиксель имеет цвет. Все цвета можно пронумеровать, а каждый номер перевести в двоичный код.

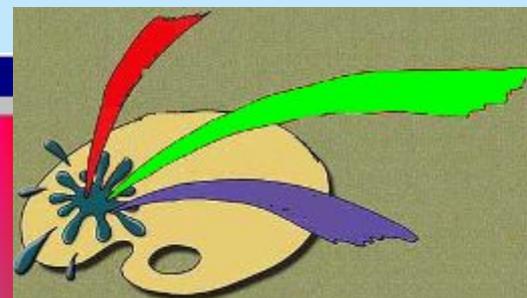
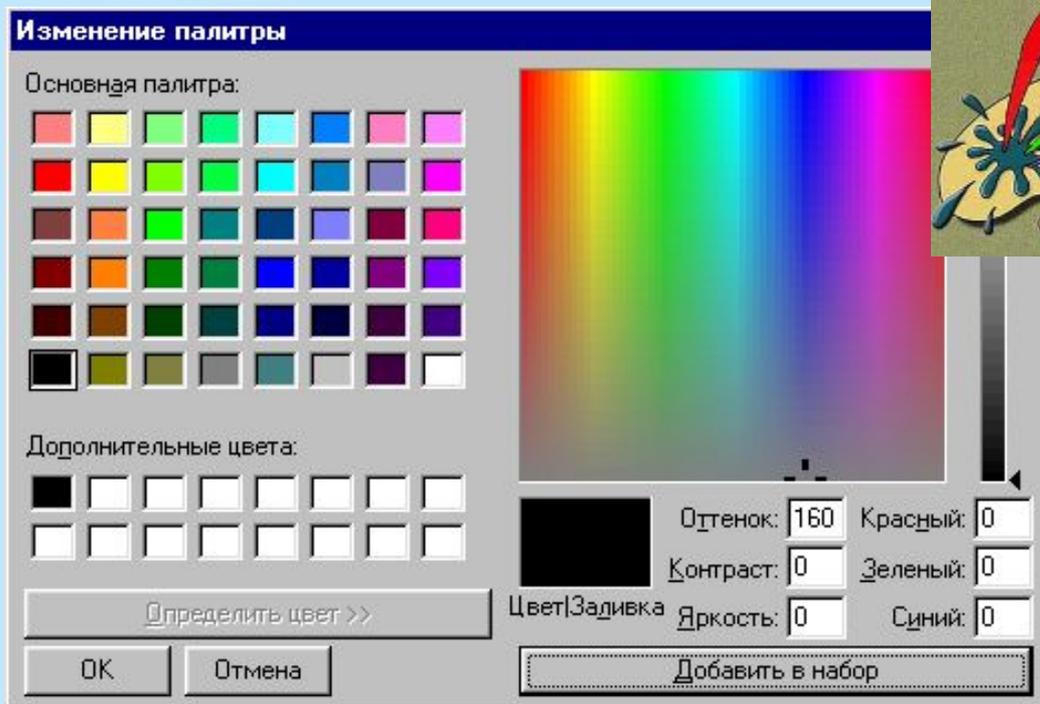


Пиксель



# Палитра

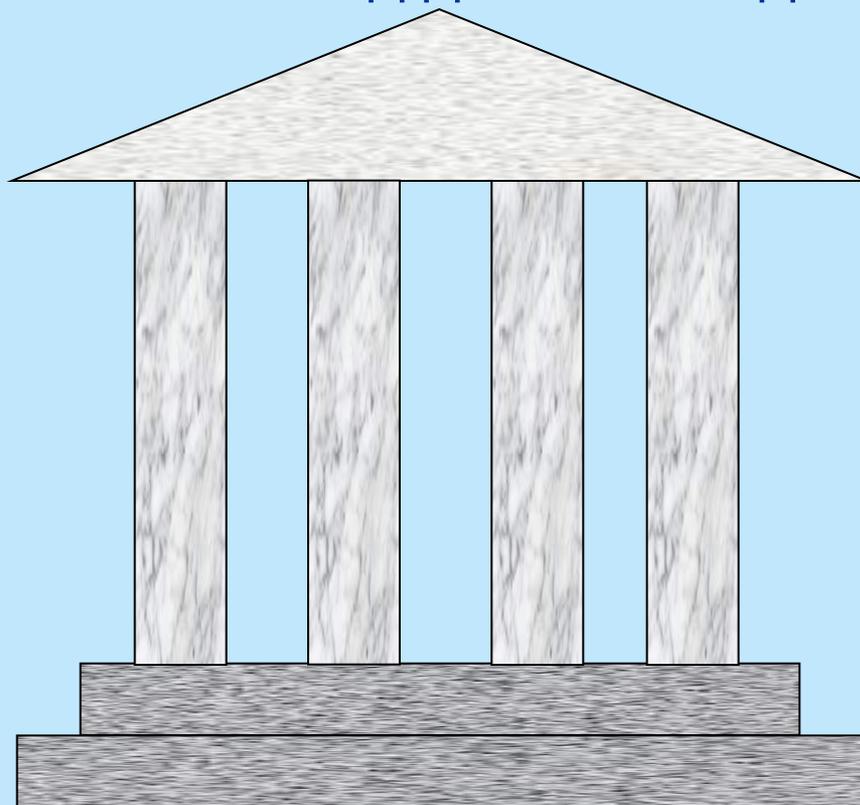
Необычайно богатая цветовая палитра современных компьютеров (более 16 миллионов оттенков) получается смешением трех основных цветов: **красного**, **зеленого** и **синего**.





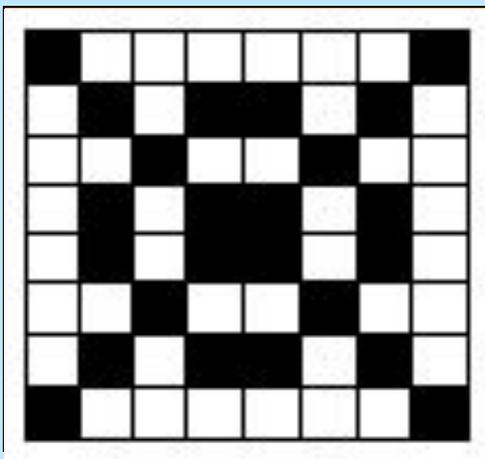
# Векторное кодирование

В графическом объекте можно выделить отдельные фрагменты – прямоугольники, треугольники, окружности, отрезки и т.д. Кодировать можно не сам рисунок, а последовательность команд для его создания.





# Самое главное



**А  
Б  
В**

**254**



**1000 0001  
0101 1010  
0010 0100  
0101 1010  
0101 1010  
0010 0100  
0101 1010  
1000 0001**



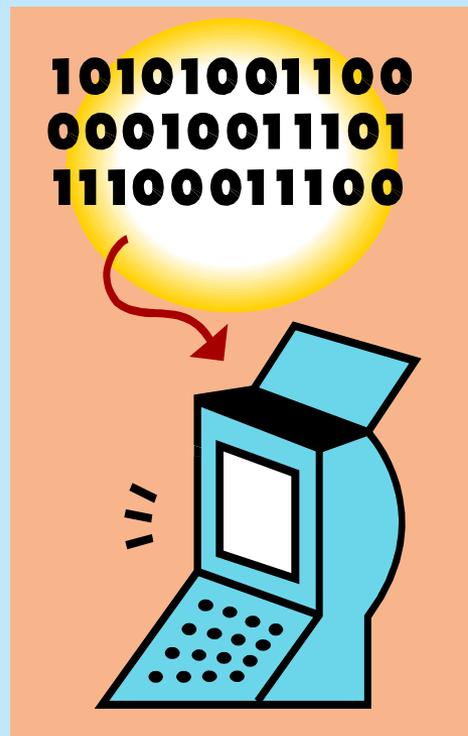
**1100 0000  
1100 0001  
1100 0010**



**1111 1110**



**1 0 1 1**



**10101001100  
00010011101  
11100011100**



# Давайте обсудим

1. Какие данные называют цифровыми?
2. Почему возникла потребность в цифровом представлении информации?
3. Как получить двоичный код целого десятичного числа?
4. Каким образом осуществляется двоичное кодирование текстовой информации?
5. Какими способами могут быть оцифрованы графические изображения?