



Белорусско-Российский университет

Кафедра «Программное обеспечение информационных технологий»

# Информатика

---

# Кодирование информации

---

КУТУЗОВ Виктор Владимирович

Республика Беларусь, Могилев, 2023

# Кодирование информации

Кодирование текстовой информации

Кодирование графической информации

- Кодирование растровых изображений
- Кодирование векторных изображений

Кодирование звуковой информации

Кодирование видеоинформации

# Основные термины

- **Данные в компьютере имеют вид кода, который состоит из единиц и нулей в разной последовательности.**
- **Код** – набор условных обозначений для представления информации
- **Код** – это правило, по которому сообщение преобразуется в цепочку знаков.
- **Кодирование** – это один из видов обработки информации. При кодировании меняется форма представления информации, а её содержание сохраняется.
- **Кодирование** – это представление информации в форме, удобной для её хранения, передачи и автоматической обработки.
- **Язык** – это система знаков и правил, используемая для записи и передачи информации.

# Примеры различных систем кодирования

С О М Р У Т Е Р

43 4F 4D 50 55 54 45 52

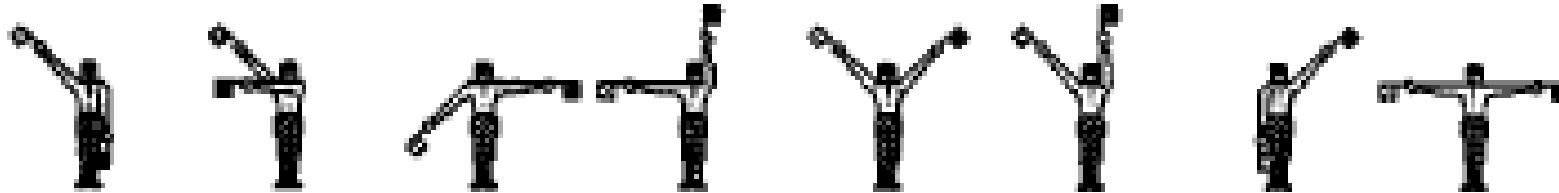
Код ASCII

— · — · — — — · — · — · — — — · — · —

Код Морзе



Код Брайля



Код морской сигнальный

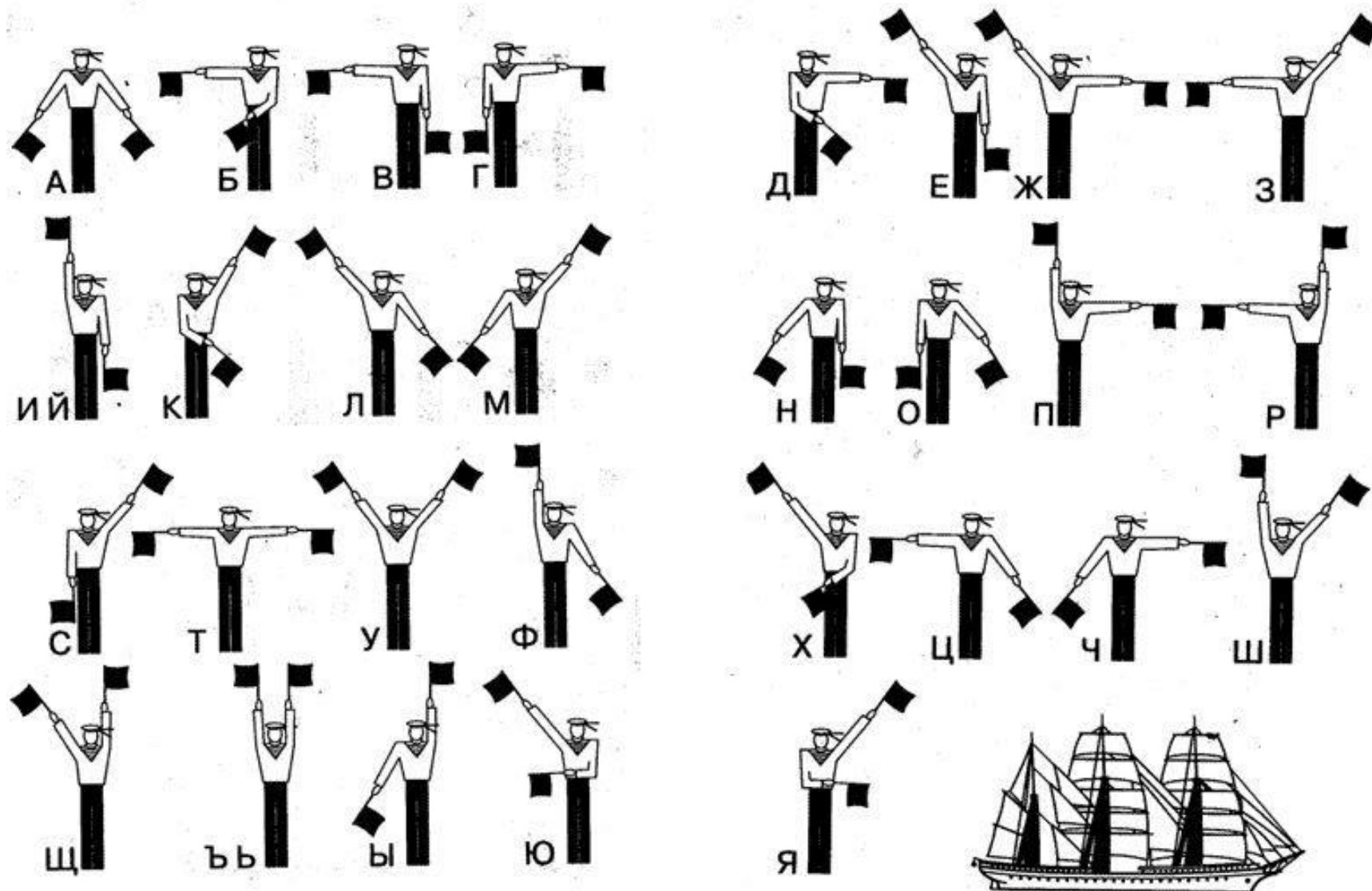
# | Код Морзе для русских букв и цифр

А	•—	О	— — —	Э	••—••
Б	—•••	П	•— —•	Ю	••— —
В	•— —	Р	•—•	Я	•—•—
Г	— —•	С	•••		
Д	—••	Т	—	1	•— — — —
Е	•	У	••—	2	••— — —
Ж	•••—	Ф	••—•	3	•••— —
З	— —••	Х	••••	4	••••—
И	••	Ц	—•—•	5	•••••
Й	•— — —	Ч	— — —•	6	—••••
К	—•—	Ш	— — — —	7	— — •••
Л	•—••	Щ	— —•—	8	— — —••
М	— —	Ь	—•• —	9	— — — —•
Н	—•	Ы	—•— —	0	— — — — —

# | Код Брайля для русских букв

<b>А</b>	<b>Б</b>	<b>В</b>	<b>Г</b>	<b>Д</b>	<b>Е</b>	<b>Ё</b>	<b>Ж</b>	<b>З</b>	<b>И</b>	<b>Й</b>
<b>К</b>	<b>Л</b>	<b>М</b>	<b>Н</b>	<b>О</b>	<b>П</b>	<b>Р</b>	<b>С</b>	<b>Т</b>	<b>У</b>	<b>Ф</b>
<b>Х</b>	<b>Ц</b>	<b>Ч</b>	<b>Ш</b>	<b>Щ</b>	<b>Ъ</b>	<b>Ы</b>	<b>Ь</b>	<b>Э</b>	<b>Ю</b>	<b>Я</b>

# Код морской сигнальный для русских букв





# Кодирование текстовой информации

# Кодирование текстовой информации

- Кодирование текстовой информации — очень распространенное явление.
- Один и тот же текст может быть закодирован в нескольких форматах.
- Принято считать, что кодирование текстовой информации появилось с приходом компьютеров. Это и так и не так одновременно.
- Кодировка в том виде, в котором мы ее знаем, действительно к нам пришла с приходом компьютеров. Но над самим процессом кодирования люди бьются уже много сотен лет. Ведь, по большому счету, **сама письменность уже является способом закодировать человеческую речь, для ее дальнейшего использования**. Вот и получается, что любая окружающая нас информация никогда не бывает представленной в чистом виде, потому что она уже каким-то образом закодирована. Но сейчас не об этом.

# Язык и алфавит

- Для того чтобы хранить и передавать информацию, её необходимо как-то закодировать, например записать с помощью знаков (символов) на каком-то языке.
- **Естественные языки** (русский, английский, китайский, немецкий, французский и многие другие) сформировались в результате развития человеческого общества и используются для общения людей.
- В большинстве **современных языков используется алфавитное письмо**, где каждый знак (или сочетание знаков) обозначает некоторый звук, так что с помощью небольшого набора знаков (алфавита) можно записать любые слова устной речи.

# Язык и алфавит



# Язык и алфавит

- **Алфавит** — это набор знаков, который используется в языке. Обычно знаки в алфавите расположены в определённом порядке.
- **Мощность алфавита** — это количество знаков в алфавите.
  - К алфавиту языка, вообще говоря, нужно отнести пробел (пропуск между словами), цифры (знаки для записи чисел), знаки препинания, скобки.
- **Сообщение** — это любой набор знаков какого-то алфавита.
- В естественных языках часто смысл слова часто можно установить только из контекста, т. е. отрывка текста, в котором оно употребляется.
- Однако есть языки специального типа, в которых каждое слово и словосочетание имеют чётко определённое единственное значение и нет никаких исключений.

# Язык и алфавит

- **Формальный язык** — это язык, в котором однозначно определяется значение каждого слова, а также правила построения предложений и придания им смысла.
- **Примеры формальных языков:**
  - Математические формулы
  - Правила записи чисел
  - Нотная запись
  - Язык записи шахматных партий
  - Алгоритмические языки
  - и так далее

# Кодирование текстовой информации

- Самый распространенный способ кодирования текстовой информации в компьютере — это ее двоичное представление, которое сплошь и рядом используется в каждом компьютере, роботе, станке и т. д.
- Все кодируется в виде слов в двоичном представлении.
- Сама технология двоичного представления информации зародилась еще задолго до появления первых компьютеров.
- Среди первых устройств, которые использовали двоичный метод кодирования, был аппарат Бодо — телеграфный аппарат, который кодировал информацию в 5 битах в двоичном представлении.
- Суть кодировки заключалась в простой последовательности электрических импульсов:  
**0 — импульс отсутствует; 1 — импульс присутствует.**

# Кодирование текстовой информации

**0 — импульс отсутствует; 1 — импульс присутствует.**

- В компьютерный мир такая кодировка пришла вместе с персонализацией самих компьютеров. То есть в первых компьютерах не было такой кодировки. Но как только компьютеры стали уходить «в массы», то резко обнаружилась потребность обрабатывать компьютерами большое количество именно текстовой информации, которую нужно было как-то кодировать. Тенденция обрабатывать большое количество текстовой информации сохранилась и в современных устройствах.
- Так получилось, что **двоичное кодирование в компьютерах связано только с двумя символами «0» и «1», которые выстраиваются в определенной логической последовательности. А сам язык подобной кодировки стал называться машинным.**

# Кодирование текстовой информации и компьютеры

- Если смотреть на текст глазами компьютера, то в тексте нет предложений, абзацев, заголовков и т. д., потому что весь текст просто состоит из отдельных символов. Причем символами будут являться не только буквы, но и цифры, и любые другие специальные знаки (+, -, \*, = и т. д.).
- Что самое интересное, даже пробелы, перенос строки и табуляция — для компьютера это тоже отдельные символы.
- Мы вводим текст в компьютер при помощи клавиатуры, символы которой мы прекрасно понимаем. Нажимая на какую-то букву, мы отправляем в оперативную память компьютера двоичное представление нажатых клавиш. Каждый отдельный символ будет представлен 8-битной кодировкой.

# **Кодирование текстовой информации и компьютеры**

- **Например буква «А» — это «11000000».**
- **Получается, что один символ — это 1 байт или 8 бит.**
- При такой кодировке, путем нехитрых подсчетов можно посчитать, что мы можем зашифровать 256 символов.
- **Для кодирования текстовой информации данного количества символов более чем предостаточно.**
- Кодирование текстовой информации в компьютерных устройствах сводится к тому, что каждому отдельному символу присваивается уникальное десятичное значение от 0 и до 255 или его эквивалент в двоичной форме от 00000000 и до 11111111.
- Люди могут различать символы по их внешнему виду, а компьютерное устройство только по их уникальному коду.

# Кодирование текстовой информации и таблицы кодировок

- Таблица кодировки — это место, где прописано какому символу какой код относится.
- Все таблицы кодировки являются согласованными — это нужно, чтобы не возникало путаницы между документами, закодированными по одной таблице, но на разных устройствах.
- **На сегодняшний день существует множество таблиц кодировок.** Из-за этого часто возникают проблемы с переносом текстовых документов между устройствами.
- Так получается, что если текстовая информация была закодирована по одной какой-то таблице, то и раскодирована она может быть только по этой таблице. Если попытаться раскодировать другой таблицей, то в результате получим только набор непонятных символов, но никак не читабельный текст.

# Таблицы кодировки

- Наиболее популярные таблицы кодировки:
  - ASCII,
  - MS-DOS,
  - ISO,
  - Windows,
  - КОИ8,
  - CP866,
  - Mac,
  - CP 1251,
  - Unicode (UTF-8, UTF-16, UTF-32),
  - и другие.

# Основные виды кодировок кириллицы

Название кодовой страницы (стандарта)	Название операционной системы (ОС)	Примечание
<b>CP866 (Code Page 866)</b>	MS DOS, OS/2	Используется в компьютерах в сеансе работы MS DOS для совместимости со старыми приложениями, для кодирования одного символа используется 8 бит.
<b>КОИ8-Р</b>	UNIX	Исторически сложившаяся кодировка, используется в русифицированных версиях операционной системы UNIX, для кодирования одного символа используется 8 бит.
<b>CP1251 (Code Page 1251)</b>	MS Windows	Кодировка, используемая в ОС Windows, для кодирования одного символа применяется 8 бит.
<b>GP10007 (Code Page 10007)</b>	Mac OS	Кодировка, используемая в ОС Macintosh фирмы Apple, для кодирования одного символа используется 8 бит.
<b>ISO-8859-5</b>	UNIX	ISO (International Standards Organization), кодировка, утвержденная в качестве стандарта русского языка

# Кодировочная таблица КОИ8-Р

—		Г	҃	҂	҄	҅	҆	҈	҉	Ҋ	ҋ	Ҍ	ҍ	Ҏ	ҏ	Ґ	ґ	Ғ	
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143				
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	°	²	•	÷
=		ҁ	е	҃	҄	҅	҆	҈	҉	Ҋ	ҋ	Ҍ	ҍ	Ҏ	ҏ	Ґ	ґ	Ғ	
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175				
		‡	Е		‡ з1	҈	҉	҈	҉	҈	҉	҈	҉	҈	҉	А	‡		‡
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191				©
ю	а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о				
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207				
п	я	р	с	т	у	ж	в	ь	ы	з	ш	э	щ	ч	ъ				
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223				
Ю	А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О				
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239				
П	Я	Р	С	Т	У	Ж	В	Ь	Ы	З	Ш	Э	Щ	Ч	Ъ				
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255				

# Кодировочная таблица ASCII

двоичный код	символ						
000 0000	[NUL]	010 0000	space	100 0000	€	110 0000	'
000 0001	[SOH]	010 0001	!	100 0001	À	110 0001	à
000 0010	[STX]	010 0010	"	100 0010	฿	110 0010	฿
000 0011	[ETX]	010 0011	#	100 0011	₵	110 0011	₵
000 0100	[EOT]	010 0100	\$	100 0100	₵	110 0100	₵
000 0101	[ENO]	010 0101	%	100 0101	₵	110 0101	₵
000 0110	[ACK]	010 0110	&	100 0110	₵	110 0110	₵
000 0111	[BEL]	010 0111	,	100 0111	₵	110 0111	₵
000 1000	[BS]	010 1000	(	100 1000	₵	110 1000	₵
000 1001	[TAB]	010 1001	)	100 1001	₵	110 1001	₵
000 1010	[LF]	010 1010	*	100 1010	₵	110 1010	₵
000 1011	[VT]	010 1011	+	100 1011	₵	110 1011	₵
000 1100	[FF]	010 1100	,	100 1100	₵	110 1100	₵
000 1101	[CR]	010 1101	-	100 1101	₵	110 1101	₵
000 1110	[SO]	010 1110	.	100 1110	₵	110 1110	₵
000 1111	[SI]	010 1111	/	100 1111	₵	110 1111	₵
001 0000	[DLE]	011 0000	0	101 0000	₱	111 0000	₱
001 0001	[DC1]	011 0001	1	101 0001	₱	111 0001	₱
001 0010	[DC2]	011 0010	2	101 0010	₱	111 0010	₱
001 0011	[DC3]	011 0011	3	101 0011	₱	111 0011	₱
001 0100	[DC4]	011 0100	4	101 0100	₱	111 0100	₱
001 0101	[NAK]	011 0101	5	101 0101	₱	111 0101	₱
001 0110	[SYN]	011 0110	6	101 0110	₱	111 0110	₱
001 0111	[ETB]	011 0111	7	101 0111	₱	111 0111	₱
001 1000	[CAN]	011 1000	8	101 1000	₱	111 1000	₱
001 1001	[EM]	011 1001	9	101 1001	₱	111 1001	₱
001 1010	[SUB]	011 1010	:	101 1010	₱	111 1010	₱
001 1011	[ESC]	011 1011	:	101 1011	₱	111 1011	₱
001 1100	[FS]	011 1100	<	101 1100	₱	111 1100	₱
001 1101	[GS]	011 1101	=	101 1101	₱	111 1101	₱
001 1110	[RS]	011 1110	>	101 1110	₱	111 1110	₱
001 1111	[US]	011 1111	?	101 1111	₱	111 1111	₱

Основная кодовая таблица ASCII		Дополнительная кодовая таблица (кириллица в так называемой «альтернативной» кодировке)	
Диапазон кодов	Набор символов	Диапазон кодов	Набор символов
00 – 1F	Управляющие символы : нуль-символ, ☺, ☻, ♥, ♦, ♣, ♠, •, □, ○, □, ♂, ♀, ↴, ↵, ↶, ↷, ▶, ◀, ↑, ↓, !!, !!, ¶, §, —, ↑, ↑, ↓, →, ←, ⊥, ↔	80 – 9F	Прописные буквы (кириллица): А, Б, В, Г, Д, Е, Ж,... Я (кроме Ё)
20 – 2F	Знаки пунктуации: пробел, !, ", #, \$, %, &, ', (,), *, +, ,, -, .., /	A0 – AF	Строчные буквы (кириллица от а до п): а, б, в, г, д, е, ж,... п (кроме буквы ё)
30 – 39	Арабские цифры: 0, 1, 2, 3, ..., 9	B0 – DF	Символы псевдографики: █, ┌, ┐, ┐ и др.
3A – 40	Знаки: . ;, <, =, >, ?, @	E0 - EF	Строчные буквы (кириллица от р до я): р, с, т, у, ф, х, ц, ...я
41 – 5A	Прописные буквы (латиница): A, B, C, D, ..., Z	F0 – F1	Буквы Ё и ё (кириллица)
5B – 60	Знаки: [, \], ^, _, `	F2 – F7	Буквы славянских алфавитов, отсутствующие в русском алфавите : Ѐ, є, І, і, Ў, ў
61 – 7A	Строчные буквы (латиница): a, b, c, d, ..., z	F8 – FF	Знаки: °, :, ., √, №, ☒, ■, ■,
7B – 7F	Знаки: {,  , }, ~, □		

# Кодировочная таблица СР1251

80	81	Г	82	,	83	Г	84	„	85	...	86	†	87	‡	88	€	89	%	8A	Љ	8B	Ќ	8C	Њ	8D	Ќ	8E	Ћ	8F	Џ
90	91	‘	92	,	93	“	94	„	95	•	96	-	97	-	98	™	99	™	9A	Љ	9B	Ќ	9C	Њ	9D	Ќ	9E	Ћ	9F	Џ
A0	A1	Ү	A2	Ү	A3	Ј	A4	Ҳ	A5	Г	A6	!	A7	՞	A8	..	A9	©	AА	Є	AВ	Ѡ	AС	–	AД	–	AЕ	®	AФ	Ї
B0	B1	±	B2	I	B3	i	B4	Г	B5	и	B6	¶	B7	·	B8	..	B9	њ	BА	Є	BВ	Ѡ	BС	j	BД	S	BЕ	S	BФ	ї
C0	C1	Б	C2	В	C3	Г	C4	Д	C5	Е	C6	Ж	C7	З	C8	И	C9	Й	CА	К	CВ	Л	CС	М	CД	Н	CЕ	О	CФ	П
D0	D1	С	D2	Т	D3	У	D4	Ф	D5	Х	D6	Ц	D7	Ч	D8	Ш	D9	Щ	DА	Љ	DВ	Ы	DС	Ђ	DД	Э	DЕ	Ю	DФ	Я
E0	E1	б	E2	в	E3	г	E4	д	E5	е	E6	ж	E7	з	E8	и	E9	й	EА	є	EВ	ѡ	EС	Ѡ	EД	н	EЕ	օ	EФ	پ
F0	F1	р	F2	с	F3	т	F4	у	F5	ф	F6	х	F7	ц	F8	ч	F9	ш	FА	ъ	FВ	ы	FС	Ђ	FД	э	FЕ	ю	FФ	я

# Таблица символов

Таблица символов

Шрифт: Arial Справка

!	"	#	\$	%	&	'	( )	*	+	,	-	.	/	0	1	2	3	4	
5	6	7	8	9	:	:	<	=	>	?	@	A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\
]	^	_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	r
q	г	с	т	у	в	w	x	y	z	{		}	~	і	џ	£	¤	¥	
ı	§	“	©	ª	«	¬	-	®	—	°	±	²	³	‘	µ	¶	·	,	1
º	»	¼	½	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Í	í
Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á
â	ã	ä	å	æ	ç	è	é	ê	ë	í	í	í	í	í	í	í	í	í	í
ö	÷	ø	ù	ú	û	ü	þ	ý	ÿ	À	â	Ã	ä	å	æ	ç	ç	ç	ç

Для копирования:  Выбрать Копировать

Дополнительные параметры

U+0021: Exclamation Mark

# Проблема разных кодировок

- Проблема использования таких различных таблиц приводила к тому, что текст, написанный на одном компьютере, мог некорректно читаться на другом.
- Например:

Любимый мир языка — это не только красивые слова и грамматика, но и различные виды кодирования. Важно понимать, что текст, написанный в одной кодировке, может не читаться корректно в другой. Например, если вы создадите документ в UTF-8 и передадите его другому человеку, который использует другой тип кодирования (например, CP-1251), то некоторые символы могут быть отображены неправильно или даже нечитаемо. Это особенно актуально для языков с диакритическими знаками, такими как русский, украинский и другие славянские языки. Для решения этой проблемы существует множество стандартов кодирования, таких как ISO-8859-1, ISO-8859-5, UTF-8 и многие другие. Каждый из них определяет способ представления символов различных языков в виде байтов.

# | Проблема разных кодировок



# Пример отображения слова в разных кодировках

- Последовательности шестнадцатеричного и двоичного кода слова «Компьютер» на основе кодировочной таблицы **CP1251** будут выглядеть следующим образом:

	К	о	м	п	ь	ю	т	е	р
CP1251	CA	EE	EC	EF	FC	FE	F2	F5	F0
CP1251	202	238	236	239	252	254	242	229	240

- Представленная кодовая последовательность в кодировках **CP866** и **КОИ8-Р** приведет к отображению следующих символов:

CP866: й ю ь я № ☐ € × Е
КОИ8-Р: й н л о э ч р е п

# Юникод - Unicode

- С конца 90-х годов проблема стандартизации символьного кодирования решается введением нового международного стандарта, который называется Unicode.
- **Юникод (англ. Unicode)** — стандарт кодирования символов.
- Полная спецификация стандарта Unicode включает в себя все существующие, вымершие и искусственно созданные алфавиты мира, а также множество математических, музыкальных, химических и прочих символов.
- **В настоящее время стандарт является преобладающим в Интернете.**

# Unicode

- **Юникод имеет несколько форм представления** (англ. Unicode transformation format, UTF):
  - **UTF-8**,
  - **UTF-16** (UTF-16BE, UTF-16LE) и
  - **UTF-32** (UTF-32BE, UTF-32LE).
- Была разработана также форма представления UTF-7 для передачи по семибитным каналам, но из-за несовместимости с ASCII она не получила распространения и не включена в стандарт. 1 апреля 2005 года были предложены две шуточные формы представления: UTF-9 и UTF-18 (RFC 4042).
- В **Microsoft Windows** NT и основанных на ней системах Windows 2000 и Windows XP в основном используется форма **UTF-16LE**.
- В **UNIX-подобных операционных системах GNU/Linux**, BSD и Mac OS X принята форма **UTF-8** для файлов и **UTF-32** или **UTF-8** для обработки символов в оперативной памяти.

# Unicode (UTF-8, UTF-16, UTF-32)

- **UTF-8** — представление Юникода, обеспечивающее наибольшую компактность и обратную совместимость с 7-битной системой ASCII; текст, состоящий только из символов с номерами меньше 128, при записи в UTF-8 превращается в обычный текст ASCII и может быть отображён любой программой, работающей с ASCII; и наоборот, текст, закодированный 7-битной ASCII может быть отображён программой, предназначеннной для работы с UTF-8. Остальные символы Юникода изображаются последовательностями длиной от 2 до 4 байт, в которых первый байт всегда имеет маску 11xxxxxx, а остальные — 10xxxxxx. В UTF-8 не используются суррогатные пары.
- **UTF-16** — кодировка, позволяющая записывать символы Юникода в диапазонах **U+0000...U+D7FF** и **U+E000...U+10FFFF** (общим количеством 1112 064). При этом каждый символ записывается одним или двумя словами (суррогатная пара). Кодировка UTF-16 описана в приложении Q к международному стандарту ISO/IEC 10646, а также ей посвящён документ IETF RFC 2781 под названием «UTF-16, an encoding of ISO 10646».
- **UTF-32** — способ представления Юникода, при котором каждый символ занимает ровно 4 байта. Главное преимущество UTF-32 перед кодировками переменной длины заключается в том, что символы Юникод в ней непосредственно индексируются, поэтому найти символ по номеру его позиции в файле можно чрезвычайно быстро, и получение любого символа n-й позиции при этом является операцией, занимающей всегда одинаковое время. Это также делает замену символов в строках UTF-32 очень простой. Напротив, кодировки с переменной длиной требуют последовательного доступа к символу n-й позиции, что может быть очень затратной по времени операцией. Главный недостаток UTF-32 — это неэффективное использование пространства, так как для хранения любого символа используется четыре байта. Символы, лежащие за пределами нулевой (базовой) плоскости кодового пространства, редко используются в большинстве текстов. Поэтому удвоение, в сравнении с UTF-16, занимаемого строками в UTF-32 пространства, зачастую не оправдано.

# Схема основной многоязычной плоскости Unicode

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

- Латинская письменность
- Нелатинские европейские письменности
- Письменности Среднего Востока и Юго-Западной Азии
- Письменности Южной и Центральной Азии
- Письменности Африки
- Письменности Восточной Азии
- Письменности Юго-Восточной Азии
- Письменности Америки
- Письменности Индонезии и Океании
- Знаки
- Системы нотописи
- Идеограммы ККЯ
- Суррогатные пары UTF-16
- Область для частного использования

По состоянию на версию Юникода 14.0

# Кодовая таблица 0400 стандарта Unicode

	0	1	2	3	4	5	6	7	8	9	А	В	С	Д	Е	F
040	È	Ё	҃	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	҃	Ќ	ѝ	Ӯ	ҏ
041	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
042	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	҆	Ы	Ь	Э	Ю	Я
043	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
044	р	с	т	у	ф	х	ц	ч	ш	щ	҆	ы	ь	э	ю	я
045	è	ë	ђ	ѓ	€	ѕ	і	ї	ј	љ	њ	Ћ	ќ	ѝ	Ӯ	ҏ
046	Ѡ	ѡ	҂	҃	҄	҅	҆	҇	҈	҉	Ҋ	ҋ	Ҍ	ҍ	Ҏ	ҏ
047	Ѱ	ѱ	Ѳ	ѳ	Ѷ	ѷ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѹ	Ѿ	Ѿ	Ѿ	ѿ
048	Ҫ	ܲ	ܳ	ܴ	ܵ	ܶ	ܷ	ܸ	ܹ	ܻ	ܼ	ܾ	ܭ	ܮ	ܰ	ܱ
049	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
04A	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
04B	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
04C	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
04D	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
04E	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ
04F	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ	Ӯ

# Таблица символов Юникода

<https://unicode-table.com/ru/>

Таблица символов Юникода

Поиск символа

Юникод Эмоджи Наборы Инструменты Алфавиты HTML-мнемоники Alt-коды »

Выпускной вечер

Популярные наборы символов

Смотреть все >

Сердечки Красивые буквы Стрелки Символы для ников Звёздочки Кавычки Символы для VK Математические знаки

0 1 2 3 4 5 6 7 8 9 A B C D E F

0000	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
0010	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
0020	!	"	#	\$	%	&	'	( )	*	+	,	-	.	/		
0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0050	P	Q	R	S	T	U	V	W	X	Y	Z	[ \ ]	^	_		
0060	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0070	p	q	r	s	t	u	v	w	x	y	z	{   }	~	DEL		
0080	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Основная латиница

Открыть на отдельной странице

Диапазон: 0000-007F

Количество символов: 128

Тип: алфавит

Языки: английский, немецкий, французский, итальянский, польский



# | Например: буква «К» в Unicode U+041A

## Техническая информация

Название в Юникоде Cyrillic Capital Letter Ka

Номер в Юникоде **U+041A**

HTML-код **&#1050;**

CSS-код **\041A**

Раздел Кириллица

Строчная **к**

Версия Юникода: 1.1 (1993)

## Свойства

Версия 1.1

Блок Кириллица

Тип парной зеркальной скобки (bidi) Нет

Композиционное исключение Нет

Изменение регистра 043A

Простое изменение регистра 043A

## Кодировка

Кодировка	hex	dec (bytes)	dec	binary
UTF-8	D0 9A	208 154	53402	11010000 10011010
UTF-16BE	04 1A	4 26	1050	00000100 00011010
UTF-16LE	1A 04	26 4	6660	00011010 00000100
UTF-32BE	00 00 04 1A	0 0 4 26	1050	00000000 00000000 00000100 00011010
UTF-32LE	1A 04 00 00	26 4 0 0	436469760	00011010 00000100 00000000 00000000



Выпускной вечер



Главная &gt; Наборы &gt; Смайлики-эмоджи «Лица»

## Смайлики-эмоджи «Лица»



U+1F600



U+1F603



U+1F604



U+1F642



U+1F643



U+1F609



U+1F618



U+1F617



U+263A



**Улыбающееся лицо с открытым ртом и смеющимися глазами >**

Номер в Юникоде: U+1F604

HTML-код: &amp;#128516;

Копировать



U+1F92A



U+1F61D



U+1F911



U+1F917



U+1F92D



U+1F92B



U+1F914



U+1F910



F605



U+1F923



U+1F602



F970



U+1F60D



U+1F929



F60B



U+1F61B



U+1F61C



Символы для ников



Символы для VK



Красивые буквы



Горячие символы



Сердечки



Забавные буквы



Символы для Facebook



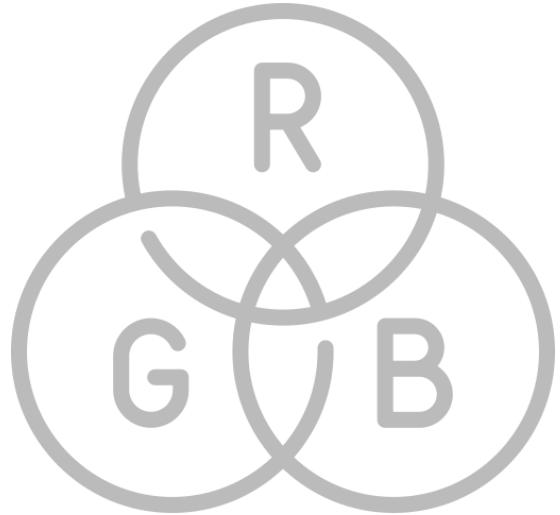
Коронавирус



Дорожные знаки



Топ-50 Эмоджи



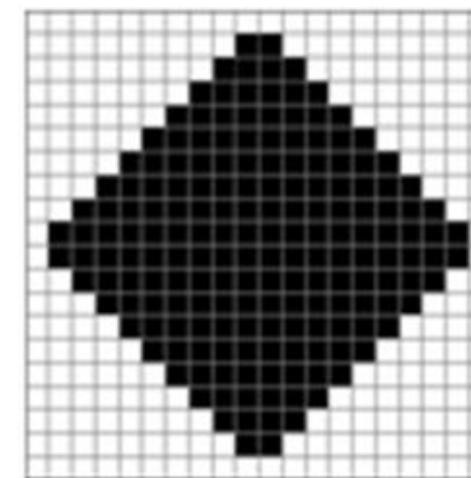
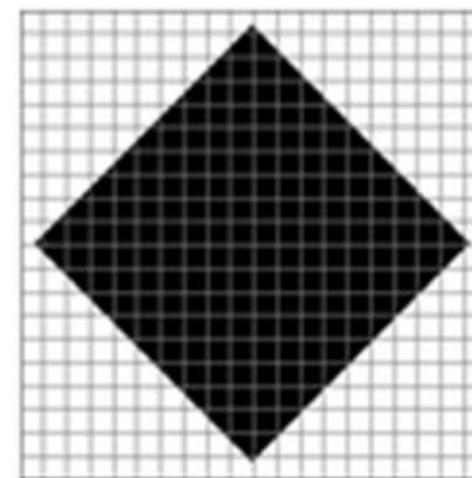
# Кодирование графической информации

# Кодирование информации

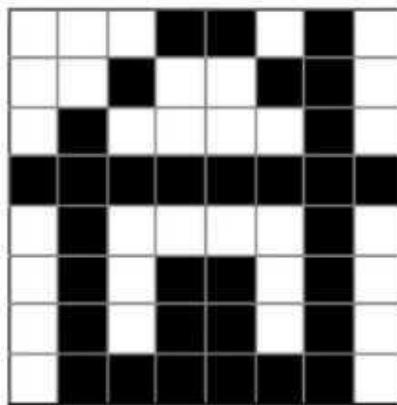
- **Графическая информация**, представленная в виде рисунков, фотографий, слайдов, подвижных изображений (анимация, видео), схем, чертежей **может создаваться и редактироваться с помощью компьютера**.
- При создании или редактировании графической информации при помощи компьютера **она** соответствующим образом **кодируется**.
- Этим занимаются **специализированные прикладные программные продукты для обработки графической информации** (преимущественно: растровой, векторной; редко: фрактальной).

# Точки. Пиксели

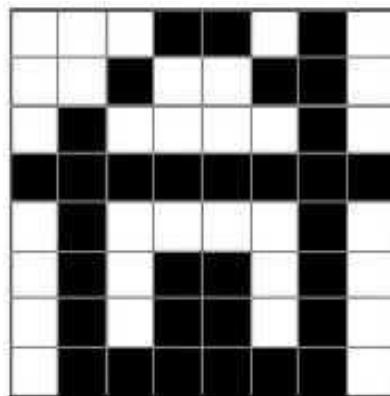
- **Файлы растровой графики предполагают хранение данных о каждой отдельной точке изображения.**
- Для отображения растровой графики не требуется сложных математических расчетов, достаточно лишь получить данные о каждой точке изображения и отобразить их на экране монитора.
- В процессе кодирования изображения производится его пространственная дискретизация, т.е. изображение разбивается на **отдельные точки**.



# Пример кодирования черно-белого рисунка



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



0	0	0	1	1	0	1	0	1A
0	0	1	0	0	1	1	0	26
0	1	0	0	0	0	1	0	42
1	1	1	1	1	1	1	1	FF
0	1	0	0	0	0	1	0	42
0	1	0	1	1	0	1	0	5A
0	1	0	1	1	0	1	0	5A
0	1	1	1	1	1	1	0	7E

0	0	0	1	1	0	1	0
1							
	A						

- Разбив рисунок на квадратики, мы выполнили его дискретизацию.
- В результате мы получили дискретный объект — набор пикселей.
- Двоичный код для чёрно-белого рисунка, полученного в результате дискретизации, можно построить следующим образом:
  - 1) кодируем белые пиксели нулями (0), а чёрные — единицами (1) ;
  - 2) записываем строки полученной таблицы одну за другой.
  - Ширина этого рисунка — 8 пикселей, поэтому каждая строка таблицы состоит из 8 двоичных разрядов — битов. Чтобы не писать очень длинную цепочку нулей и единиц, удобно использовать шестнадцатеричную систему счисления, закодировав 4 соседних бита (тетраду) одной шестнадцатеричной цифрой.
  - Например, для первой строки получаем код  $1A_{16}$ .
  - Для всего рисунка:  $1A2642FF425A5A7E_{16}$ .

# Пример кодирования черно-белого рисунка

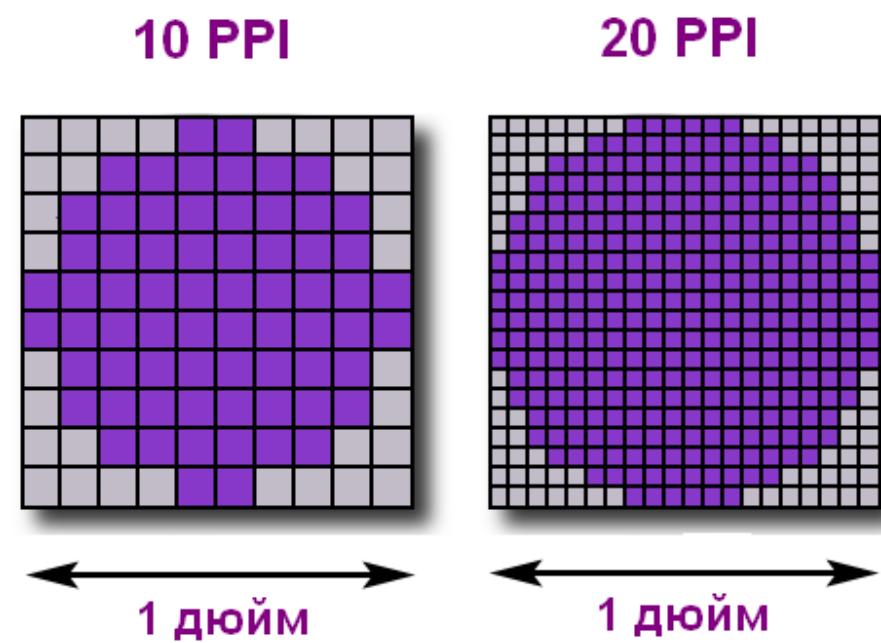
Образ символа								Коды строк матрицы	
0	1	2	3	4	5	6	7	0000 0000	00 <sub>h</sub>
								0000 1110	0E <sub>h</sub>
								0001 0010	12 <sub>h</sub>
								0010 0010	22 <sub>h</sub>
								0011 1110	3E <sub>h</sub>
								0010 0010	22 <sub>h</sub>
								0010 0010	22 <sub>h</sub>
								0010 0010	22 <sub>h</sub>
7 6 5 4 3 2 1 0									

# Пиксель

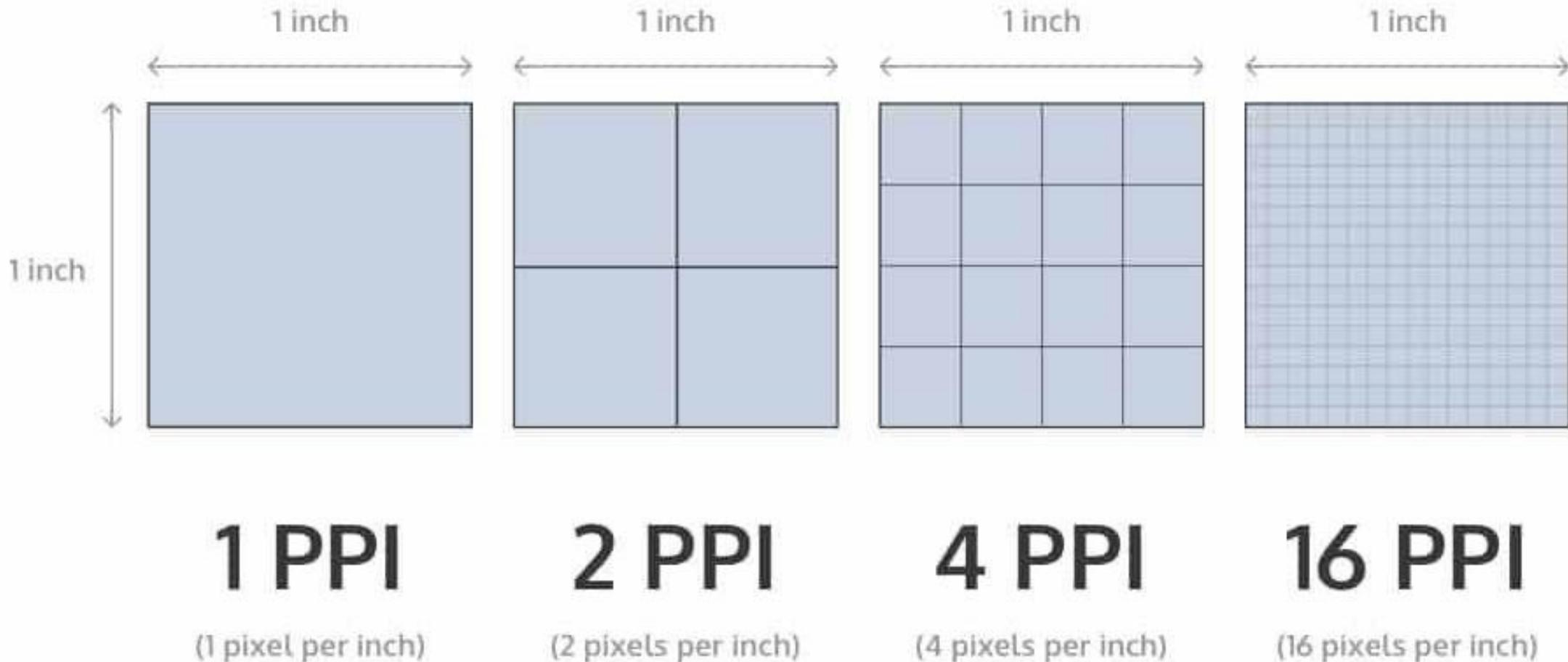
- **Пиксель** (англ. pixel = picture element, элемент рисунка) – это наименьший элемент рисунка, для которого можно задать свой цвет.
- **Чем больше пикселей на единицу площади содержит изображение, тем более оно детально.**
- **Качество растрового изображения** определяется двумя основными параметрами — **разрешением** (количеством точек по горизонтали и вертикали) **и используемой палитрой цветов** (набором цветов, при этом каждая точка может иметь количество цветов, равное 16, 256, 65 536 и т.д.). Разрешение задается указанием числа точек по горизонтали на число точек по вертикали. Например, 1024 на 768 точек.

# Разрешение - dpi, ppi

- **Разрешение** – это количество пикселей, приходящихся на единицу линейного размера изображения.
- Для обозначения разрешающей способности различных процессов преобразования изображений (сканирование, печать, растеризация и т. п.) чаще всего используют следующие термины:
  - **dpi** (англ. dots per inch)  
**количество точек на дюйм.**
  - **ppi** (англ. pixels per inch)  
**количество (плотность) пикселей на дюйм.**



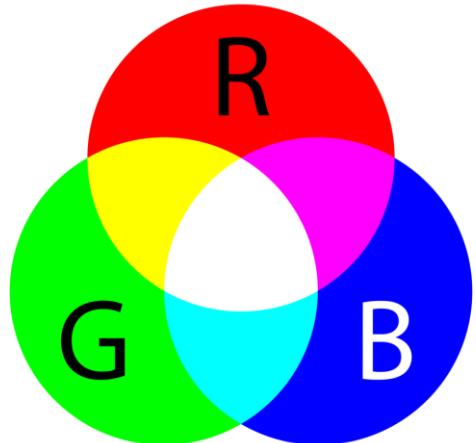
# PPI (pixels per inch) количество (плотность) пикселей на дюйм



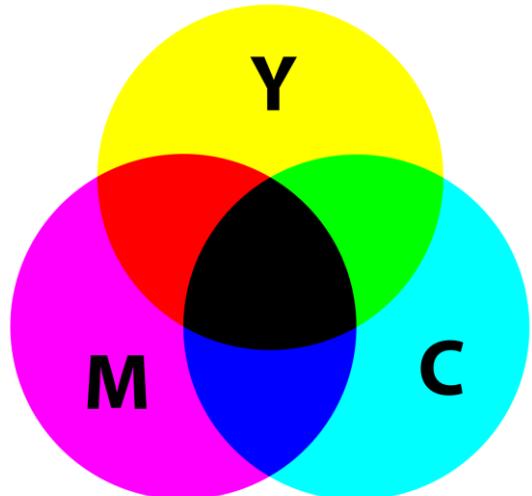
# Кодирование цвета

- При кодировании цветных изображений изображение разбивается на отдельные точки с указанием цвета.
- **Каждый пиксель растрового цветного изображения — объект, характеризуемый определённым цветом, яркостью и, возможно, прозрачностью.** Один пиксель может хранить информацию только об одном цвете, который и ассоциируется с ним.
- **Яркость** каждой составляющей обычно кодируется целым числом от 0 до 255. При этом код цвета – это тройка чисел (R,G,B), яркости отдельных каналов.
- **Цвет (0,0,0)** – это черный цвет, а (255,255,255) – белый. Или десятичной дробью от 0 до 1. При кодировании используется специализированная цветовая модель.
- **Глубина цвета** – это количество битов, используемых для кодирования цвета пикселя.

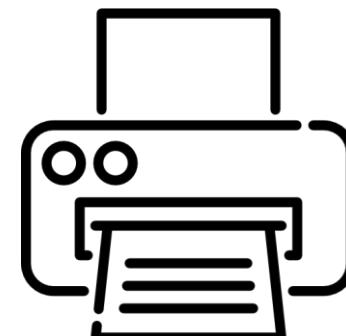
# Цветовые модели RGB, CMYK



- Цветовая модель **RGB** (**R** - красный, **Green** - зеленый, **Blue** - синий) лучше всего описывает цвет, который излучается некоторым устройством, например, экраном компьютера, ноутбука, планшета или смартфона

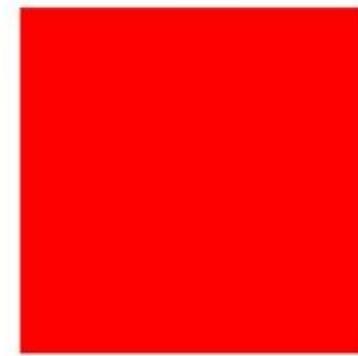


- Цветовая модель **CMYK** (**Cyan** - голубой, **Magenta** - пурпурный, **Yellow** - жёлтый, **black** - черный), часто применяется для вывода рисунков на печать.

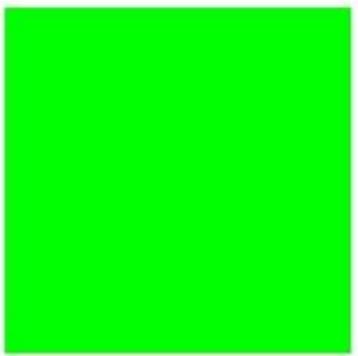




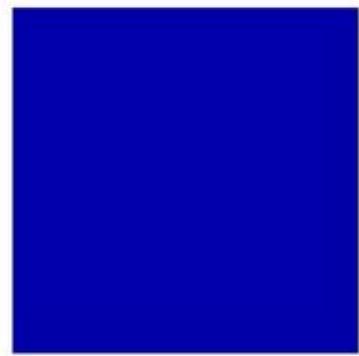
# RGB



(R) Red



(G) Green



(B) Blue



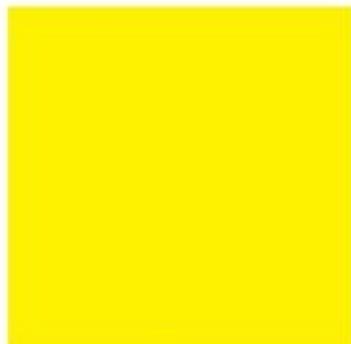
# CMYK



(C) Cyan



(M) Magenta

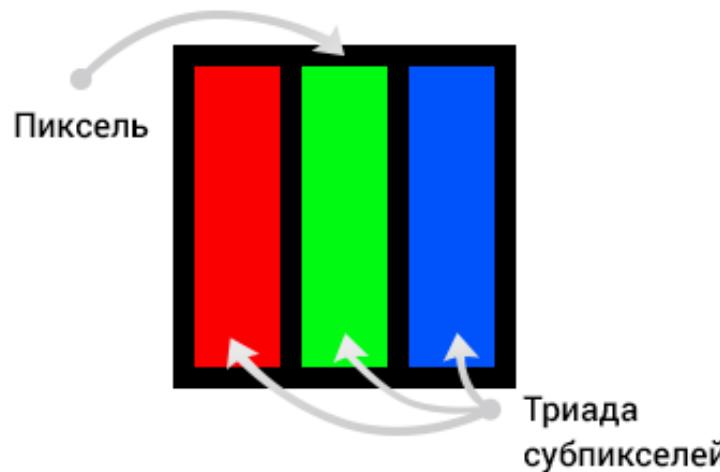
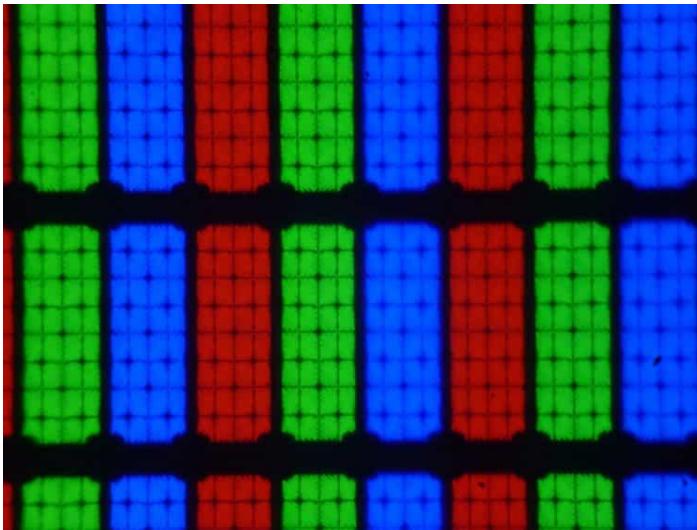


(Y) Yellow



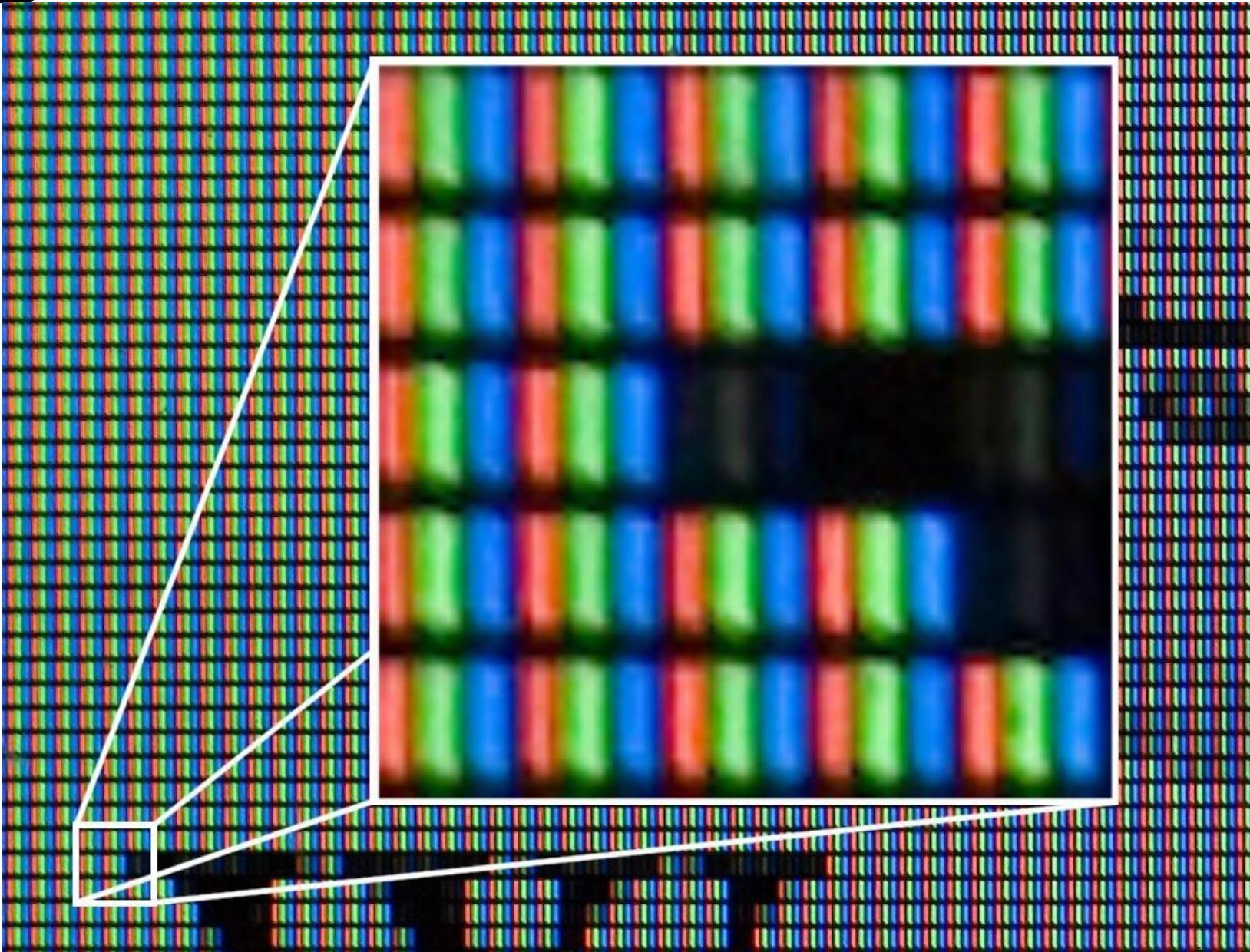
(B) Black

# Пиксель

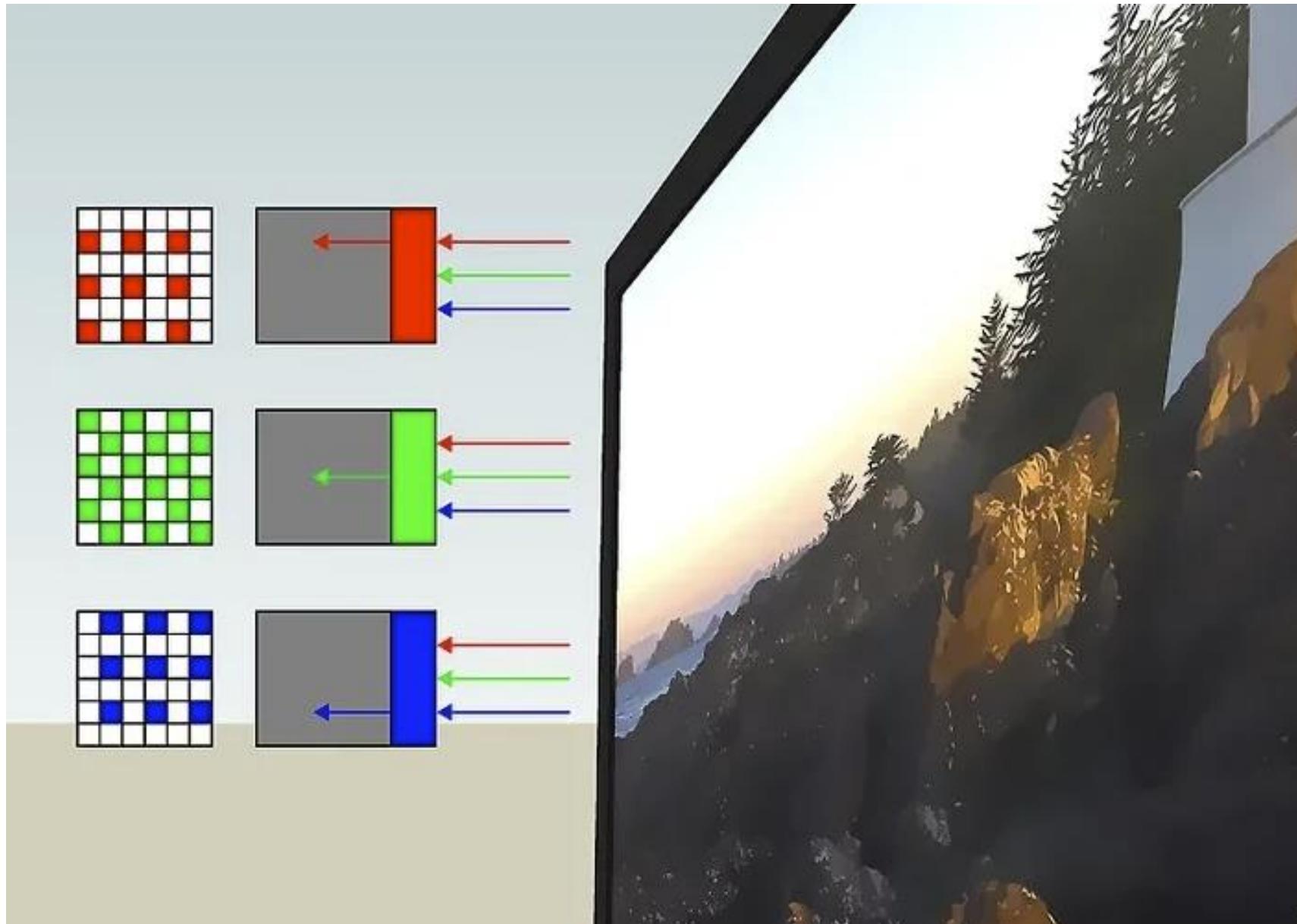


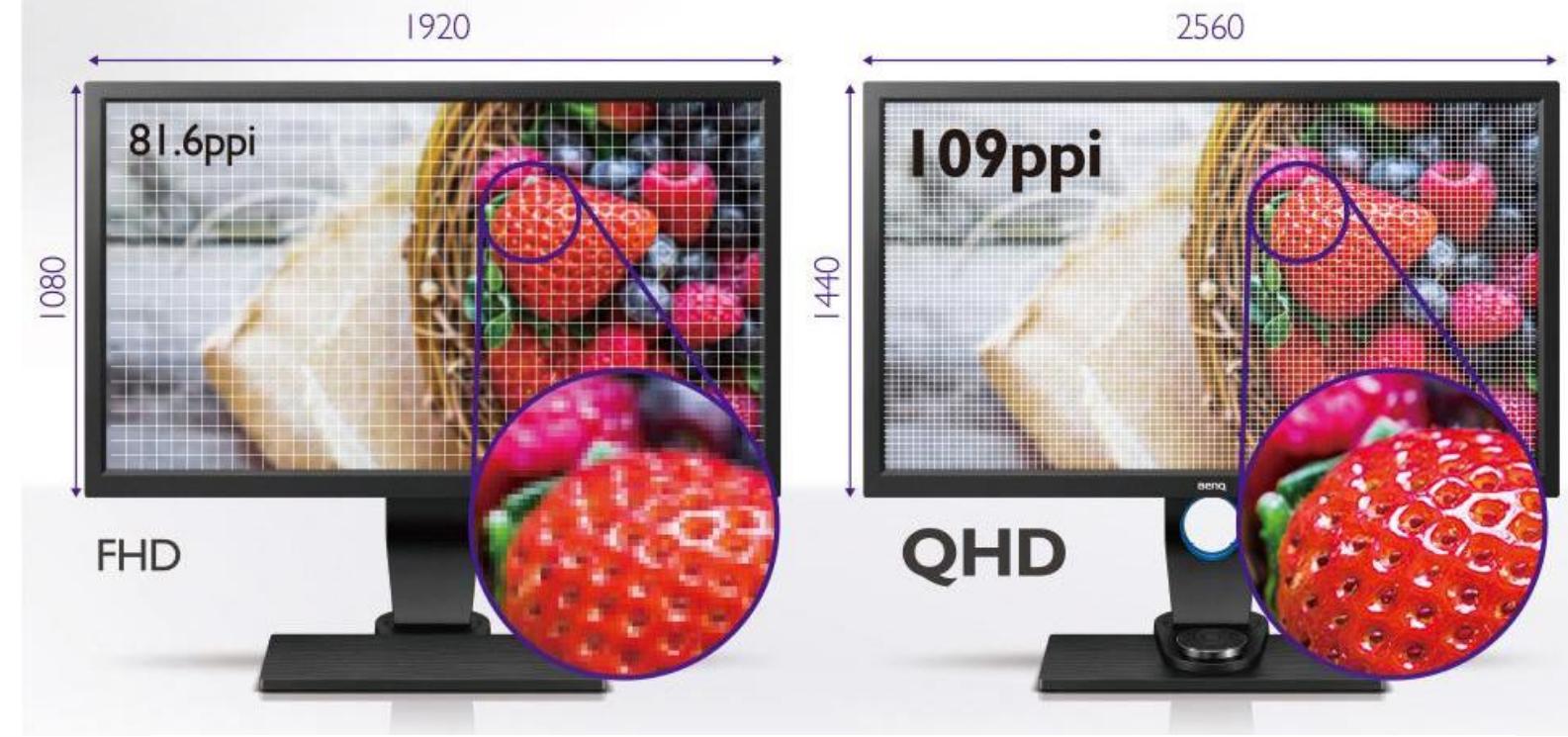
- По своей сути **пиксель** – это небольшая точка жидкокристаллического монитора, отвечающая за показ изображения.
- Такая точка содержит в себе три части, именуемые **субпикселями**, которые могут обладать красной, синей или зелёным цветом.
- Именно пиксели и создают на жидкокристаллическом мониторе полное изображение и позволяют владельцам компьютера или телевизора смотреть фильмы, играть в компьютерные игры и работать на ПК.

# Пиксель

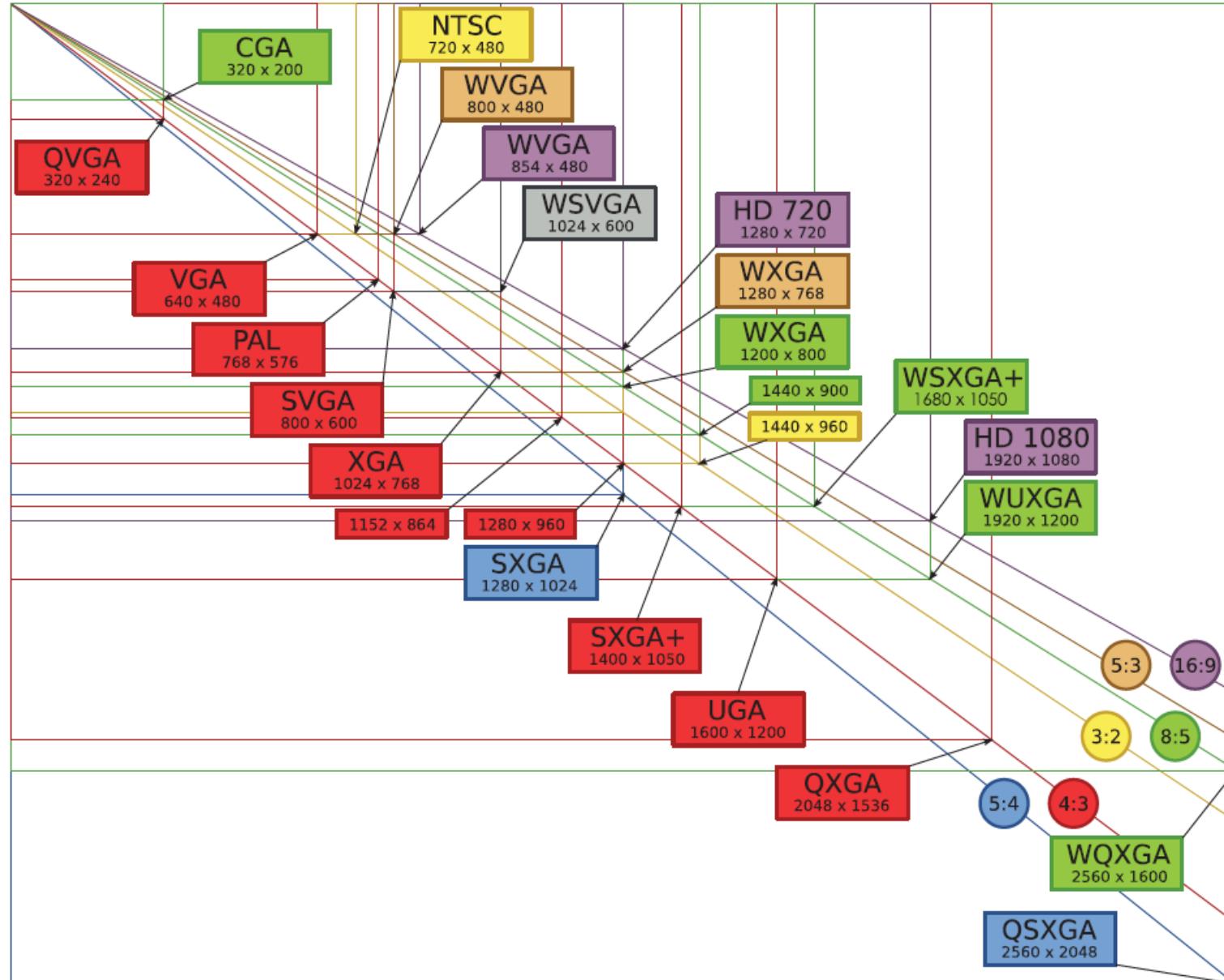


# Пиксель





# Разрешение изображения

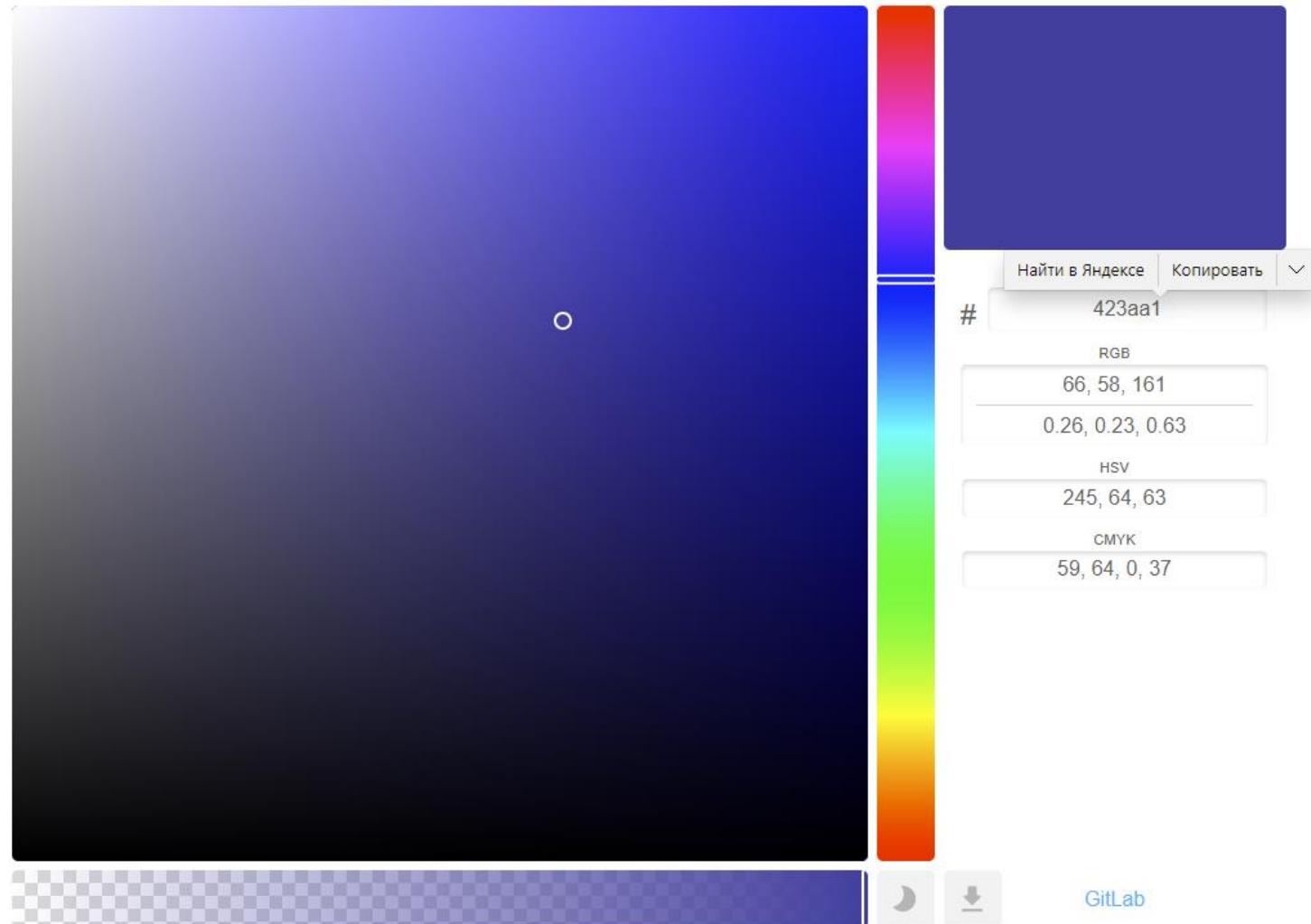


# RGB цвета и их кодирование

Color	HTML / CSS Name	Hex Code #RRGGBB	Decimal Code (R,G,B)
Black	Black	#000000	(0,0,0)
	White	#FFFFFF	(255,255,255)
Red	Red	#FF0000	(255,0,0)
Lime	Lime	#00FF00	(0,255,0)
Blue	Blue	#0000FF	(0,0,255)
Yellow	Yellow	#FFFF00	(255,255,0)
Cyan / Aqua	Cyan / Aqua	#00FFFF	(0,255,255)
Magenta / Fuchsia	Magenta / Fuchsia	#FF00FF	(255,0,255)
Silver	Silver	#C0C0C0	(192,192,192)
Gray	Gray	#808080	(128,128,128)
Maroon	Maroon	#800000	(128,0,0)
Olive	Olive	#808000	(128,128,0)
Green	Green	#008000	(0,128,0)
Purple	Purple	#800080	(128,0,128)
Teal	Teal	#008080	(0,128,128)
Navy	Navy	#000080	(0,0,128)

# Конвертер цветов <https://csscolor.ru/>

HTML CSS Палитра цветов



HTML CSS Палитра цветов помогает подобрать цвет в виде HEX, RGB, RGBA, HSV и CMYK записи цветовой модели.

# Конвертор цветов

<https://colorscheme.ru/color-converter.html>

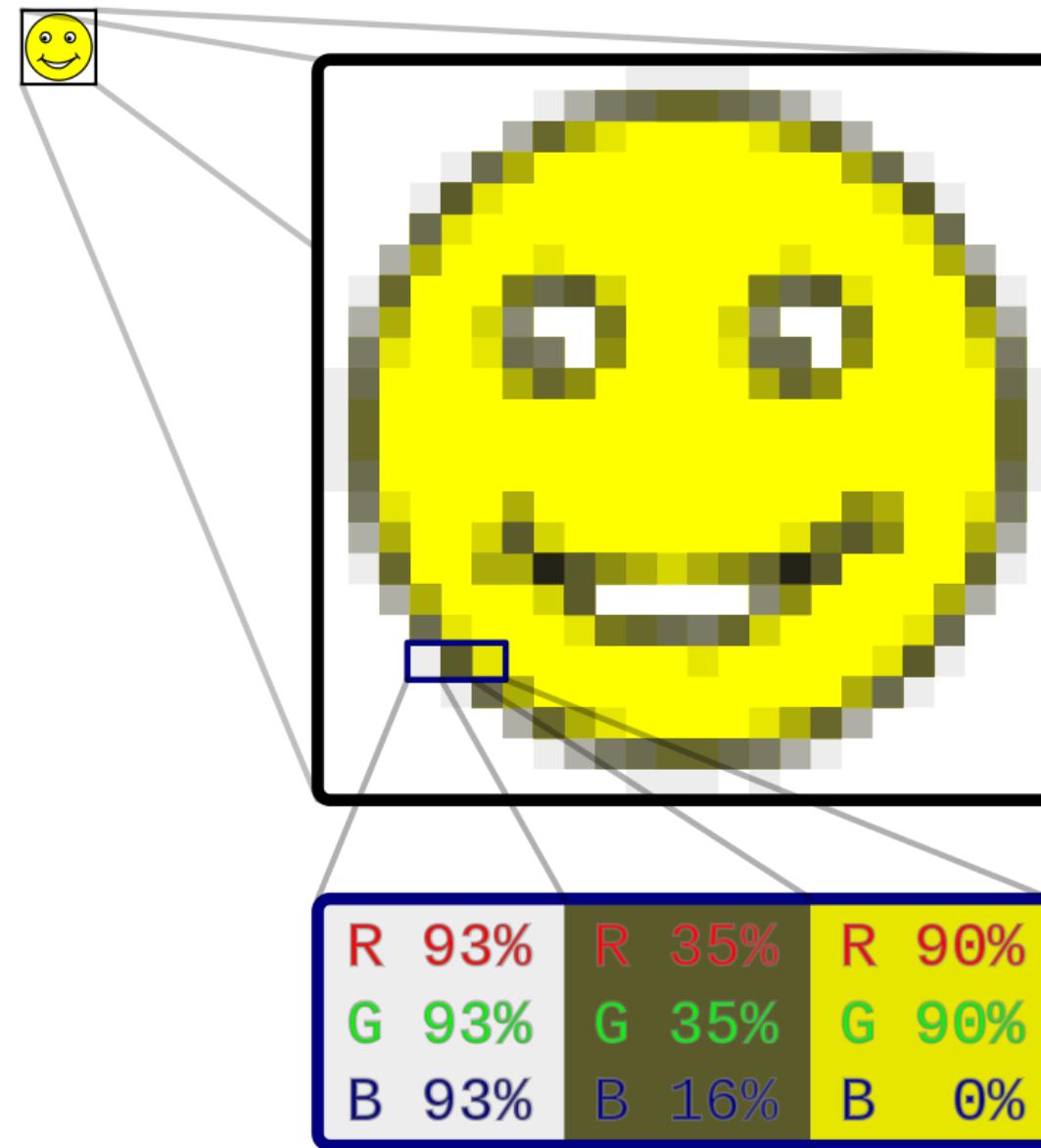
The screenshot shows the 'ColorScheme.Ru' website interface. At the top, there's a navigation bar with tabs: 'Карта Сайта', 'Названия Цветов', 'Цвета Автомобилей', 'Имена цветов в HTML', and 'Конвертер Цветов'. The 'Конвертер Цветов' tab is active. Below the navigation, the title 'КОНВЕРТЕР ЦВЕТОВ' is displayed. A descriptive text explains that the converter allows users to easily convert colors between various color models. It highlights support for CSS3 color formats like HEX, RGB, HSL, CMYK, XYZ, and LAB, as well as conversion from HEX to RGB/RGBA and HSL/HSLA to RGB/RGBA. A note also mentions a special feature for alpha channel replacement.

Цветовая модель:  # ba0404

Цвет фона для заполнения альфа-канала (RGBA в RGB)  
0 Точность округления (кол-во знаков после запятой)

HEX	#ba0404	WEBSAFE	#C00
RGB	rgb(186, 4, 4);	RGBA	rgba(186, 4, 4, 1);
% RGB	rgb(73%, 2%, 2%);	% RGBA	rgb(73%, 2%, 2%, 1);
HSL	hsl(0, 96%, 37%);	HSLA	hsla(0, 96%, 37%, 1);
CMYK	0%, 98%, 98%, 27%	HSB/HSV	0, 98%, 73%
XYZ	20, 11, 1	LAB	40, 58, 54

# Растровое изображение



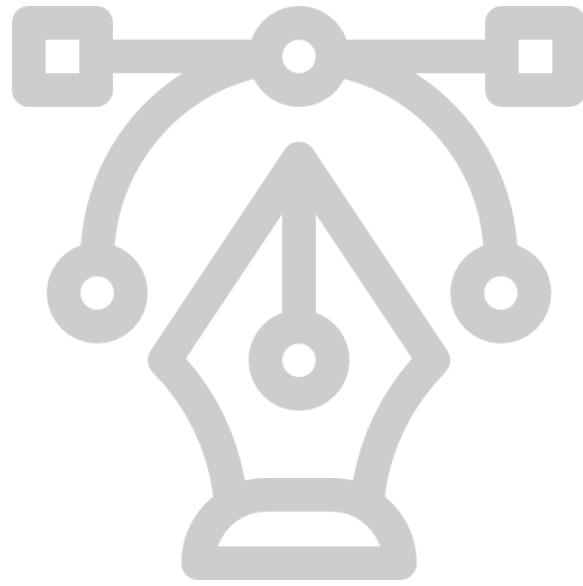
# Растровое кодирование изображений

## • Достоинства

- это универсальный метод (можно закодировать любое изображение);
- это единственный метод для кодирования и обработки размытых изображений, не имеющих четких границ, например, фотографий.
- Высокая скорость обработки сложных изображений, если не нужно масштабирование;
- Растровое представление изображения естественно для большинства устройств ввода-вывода графической информации, таких как мониторы (за исключением векторных устройств вывода), матричные и струйные принтеры, цифровые фотоаппараты, сканеры, а также сотовые телефоны.

## • Недостатки

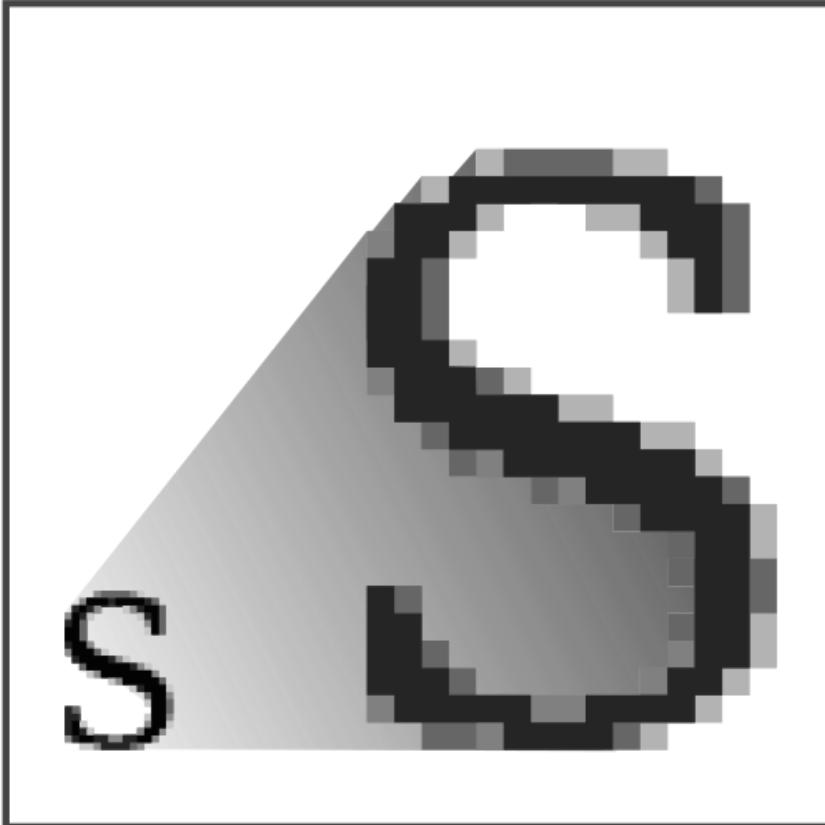
- при дискретизации всегда есть потеря информации;
- при изменении размеров изображения искажается цвет и форма объектов на рисунке;
- размер файла определяется только разрешением и глубиной цвета; как правило, растровые рисунки имеют большой объем.



# Кодирование графической информации

## Векторные изображения

# Растровые и Векторные изображения

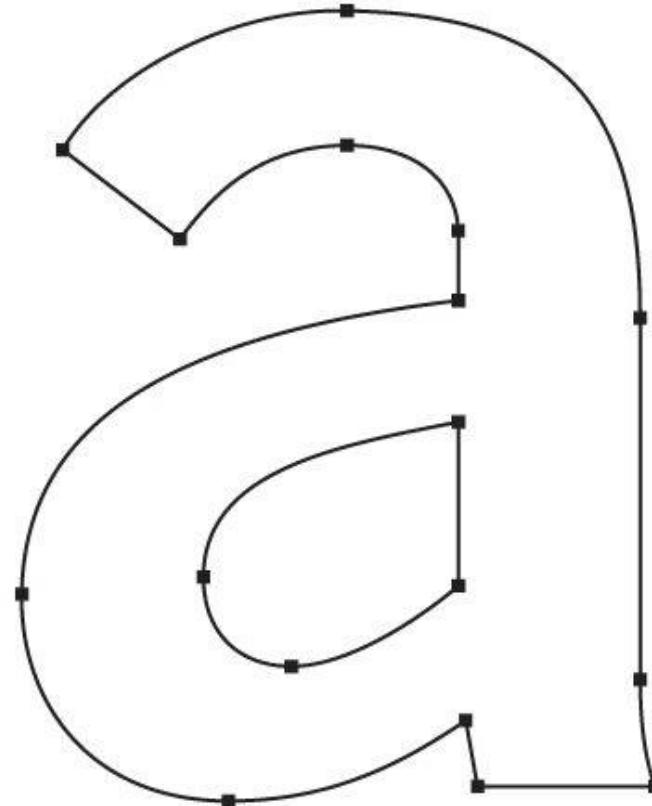


РАСТР  
.jpeg .gif .png

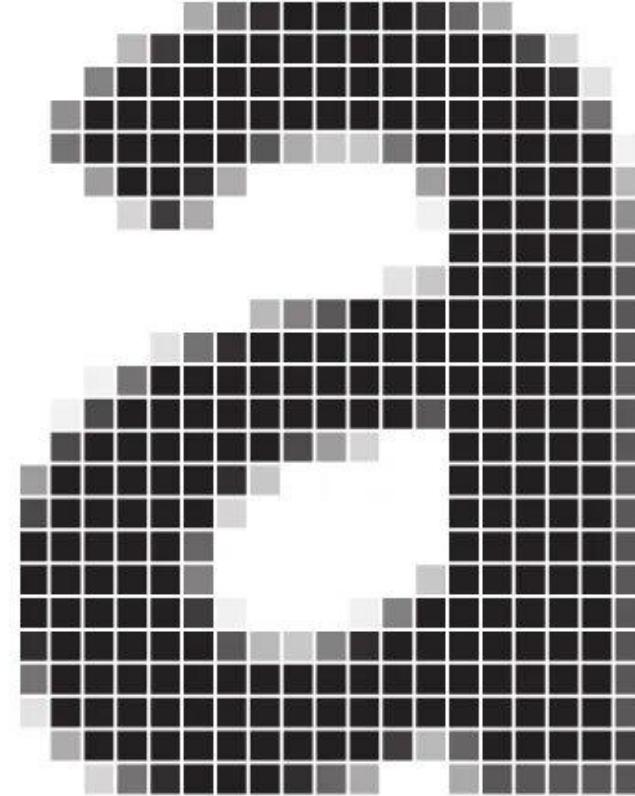


ВЕКТОР  
.svg

# Растровые и Векторные изображения

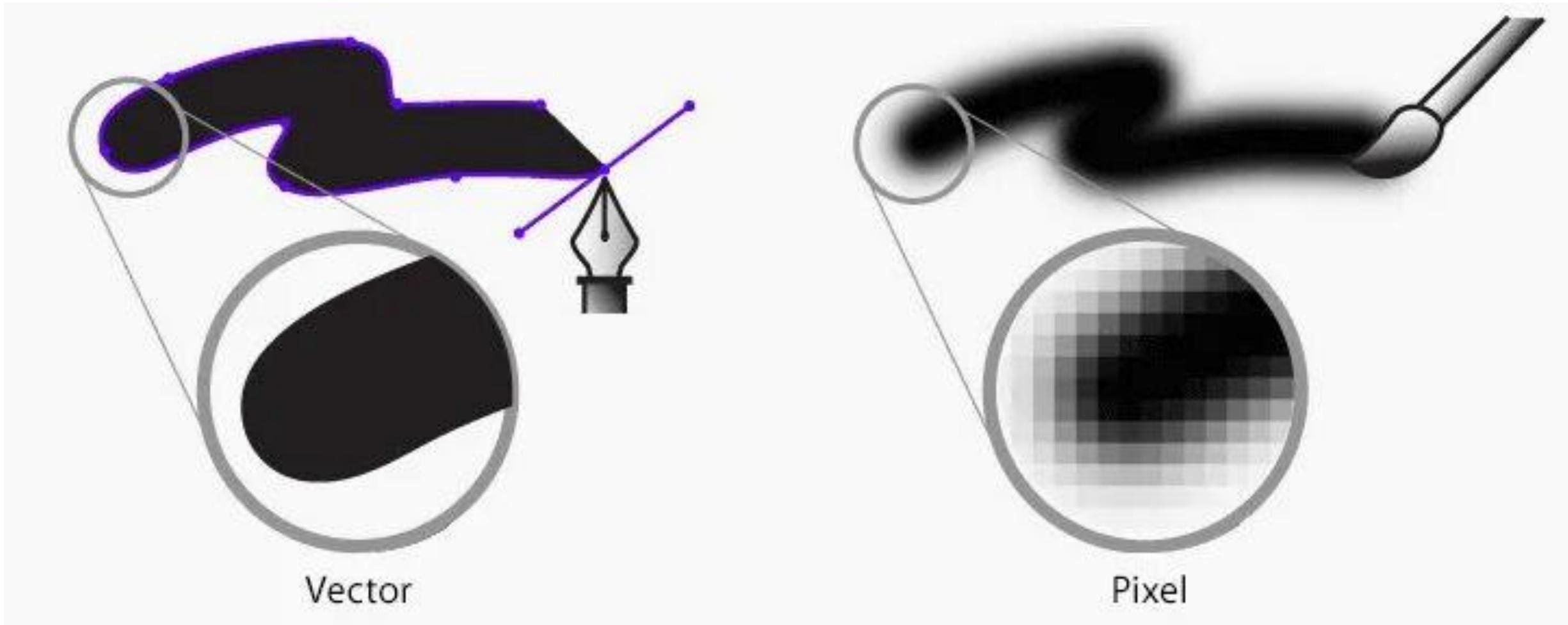


VECTOR



RASTER

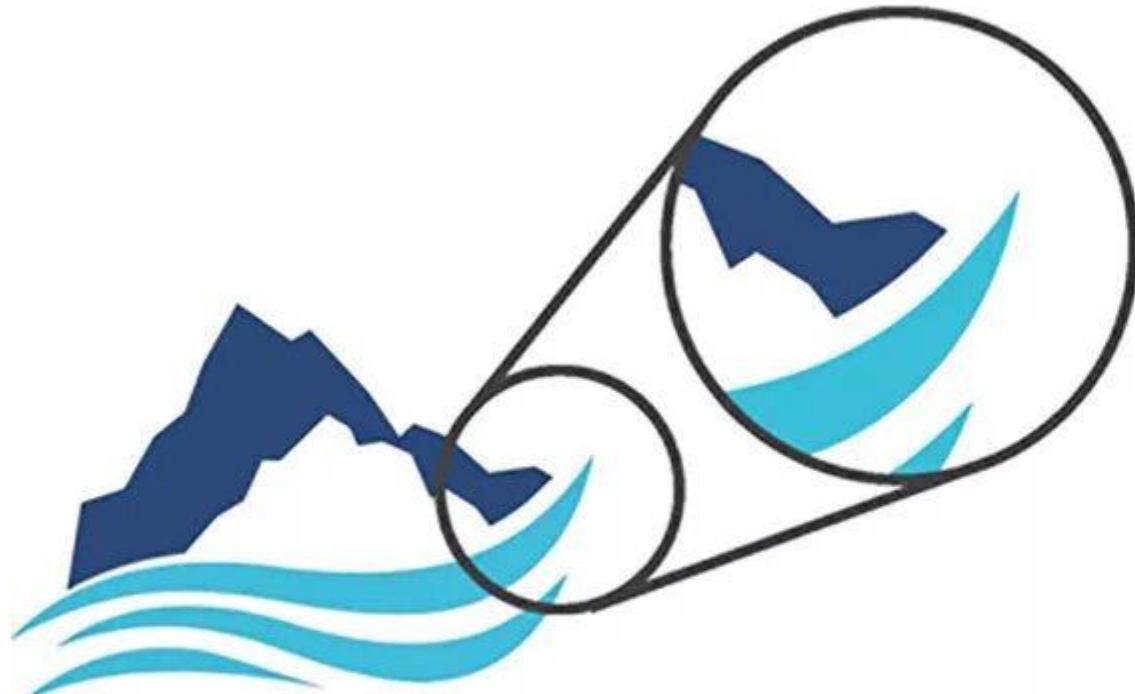
# Растровые и Векторные изображения



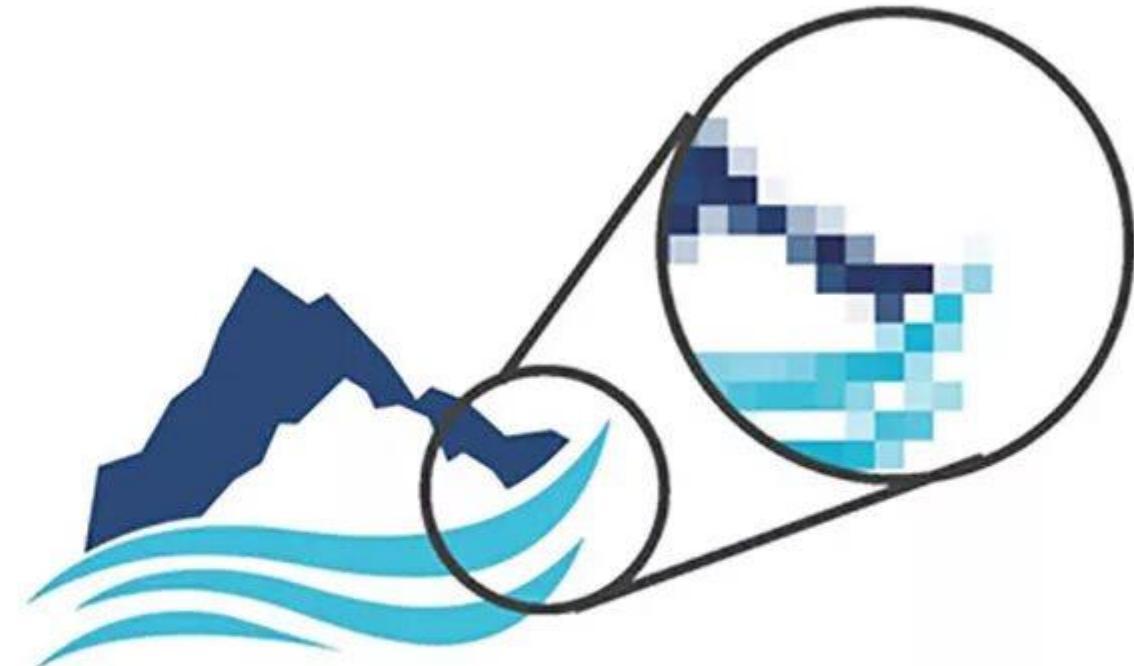
Vector

Pixel

# Растровые и Векторные изображения



Векторная графика



Растровая графика

# Векторные изображения

- Особняком идут графические форматы файлов (и работающие с ними программы), сохраняющие векторную графическую информацию.
- В противовес хранению растра, указанные форматы хранят изображения не в виде точек, представляющих собой сетку, а в виде некоторых легко масштабируемых примитивов (линии, дуги, прямоугольники, кривые Безье и пр.).
- По сути изображение состоит из кусочков ломаных и вычисляется в процессе отображения. Это основное отличие векторной графики от растровой.
  - Например, в таком формате оптимально хранить масштабируемые шрифты (TrueType, OpenType58 и др.). С 2016 года используется новый стандарт OpenType-SVG представления цветных шрифтов.

# Векторные изображения

- **Векторный рисунок** хранится в памяти как множество геометрических фигур с заданными свойствами контура и заливки внутренней области. Для каждой из фигур задаются координаты на рисунке, размеры, цвет и стиль контура и заливки. Поэтому при изменении размеров такой рисунок не искажается, а просто увеличивается или уменьшается.
- **Векторный рисунок хранит в памяти компьютера не отдельные пиксели, а информацию о геометрических фигурах, из которых составлен рисунок:**
  - толщину, цвет и стиль контура;
  - стиль заливки (один цвет, переход между несколькими цветами, узор);
  - координаты фигуры, угол поворота, угол наклона.

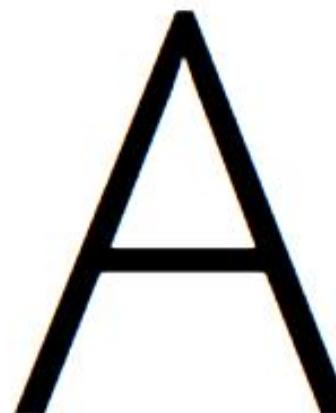
# Векторные изображения

- Векторное кодирование применяют для хранения схем, чертежей, карт, планов.
- Для создания и редактирования векторных рисунков используют векторные редакторы (Corel Draw, Adobe Illustrator и др.).



# Пример векторных изображений

- Применение векторной графики в шрифтах (примеры изображения букв взяты с сайта <http://colorfonts.wtf> , слева направо: обычным (чёрно-белый) шрифтом, цветным векторным шрифтом, цветным векторно-растровым шрифтом)



Standard vector font



Color vector font



Color bitmap font

# Достоинства и Недостатки

- **Достоинства :**

- нет потери информации
- при увеличении или уменьшении рисунки не искажаются
- рисунки занимают немного места в памяти

- **Недостатки :**

- очень сложно (и не нужно) кодировать так изображения изображения без четких границ объектов
- не подходит для кодирования и обработки фотографий фотографий



Кодирование  
звуковой  
информации

# Звук

- **Звук – это колебания физической среды.** В повседневной жизни такой средой является воздух. **Чаще всего звуковые колебания преобразуют в электрические с помощью микрофона.** Представление о форме этого сигнала можно получить через программу Windows Player или аналогичные.

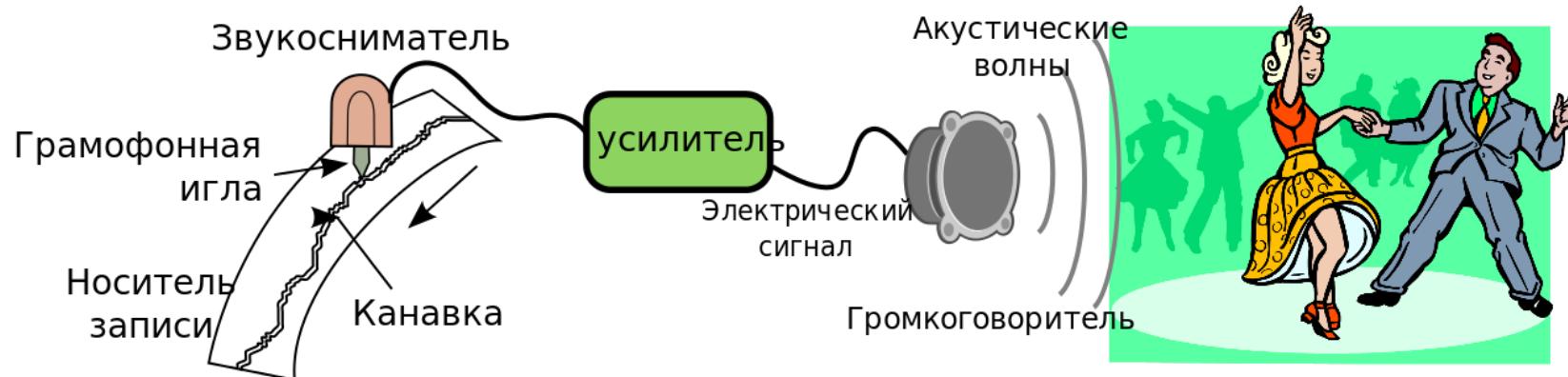


# Пример записи и воспроизведения звука

## Звукозапись

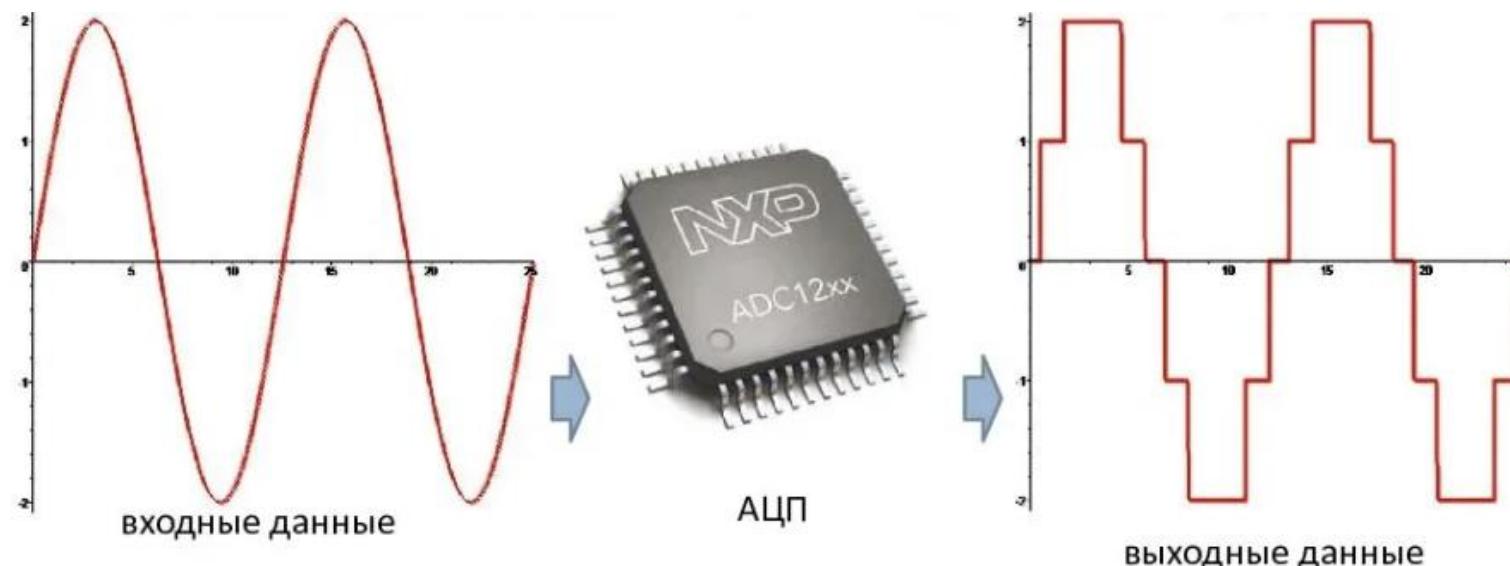


## Воспроизведение



# Звук

- Звуковой (аудио) сигнал имеет аналоговую природу.
- Для того чтобы преобразовать его в дискретную форму используют специальный блок, входящий в состав звуковой карты компьютера, **АЦП (аналого-цифровой преобразователь)**. Основной принцип его работы заключается в том, что интенсивность звукового сигнала фиксируется не непрерывно, а периодически, в определенные моменты времени.



# Звук

- Частоту, характеризующую периодичность измерения, называют **частотой дискретизации**. Считается, что для хорошего воспроизведения звука она должна, по крайней мере, в два раза превышать максимальную частоту волны, входящей в спектр звукового сигнала.
- Человеческое ухо воспринимает как звук колебания в диапазоне частот до **22 000 Гц**. Следовательно, для хорошего воспроизведения музыки частота дискретизации должна быть не менее **44 000 Гц**.
- При записи речи такое высокое качество воспроизведения не нужно. Определено, что речь воспринимается вполне разборчиво уже при частоте дискретизации **8 000 Гц**.

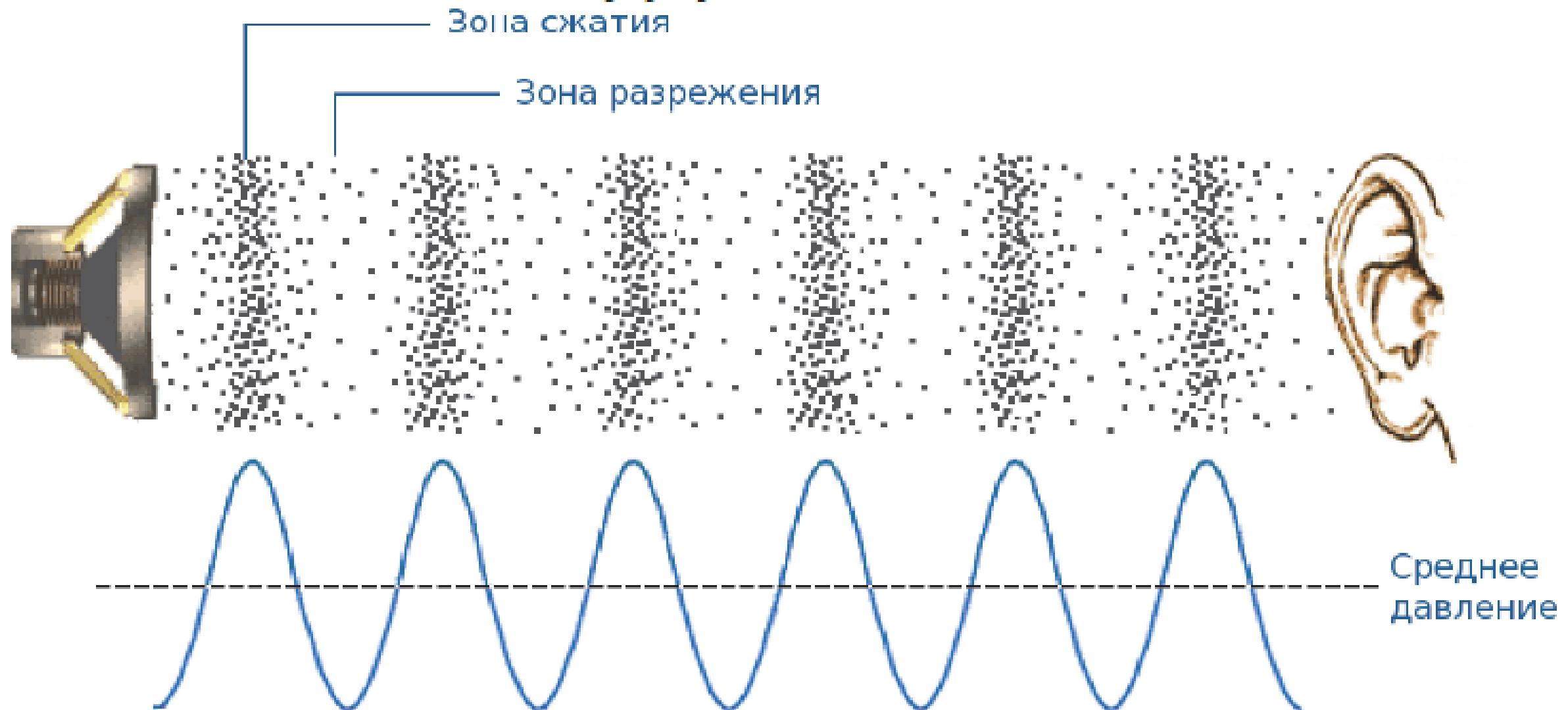
# Звук

- Помимо дискретизации по времени АЦП проводит дискретизацию и по интенсивности звука, т. е. по амплитуде звукового сигнала. В АЦП закладывается сетка стандартных интенсивностей – глубина кодирования (256 или 65 536 уровней), и реальная интенсивность округляется до уровня, ближайшего по сетке.
- Обратное преобразование закодированного таким образом звука в аналоговую форму, воспринимаемую человеческим ухом, производится блоком ЦАП (цифро-аналоговый преобразователь).
- По закодированным точкам время-интенсивность с помощью интерполяции рассчитывается гладкая непрерывная кривая, которая используется при восстановлении звукового сигнала.
- Для проведения расчетов, восстанавливающих вид звукового сигнала, выпускаются специализированные микропроцессоры, DSP (Digital Signal Processor).

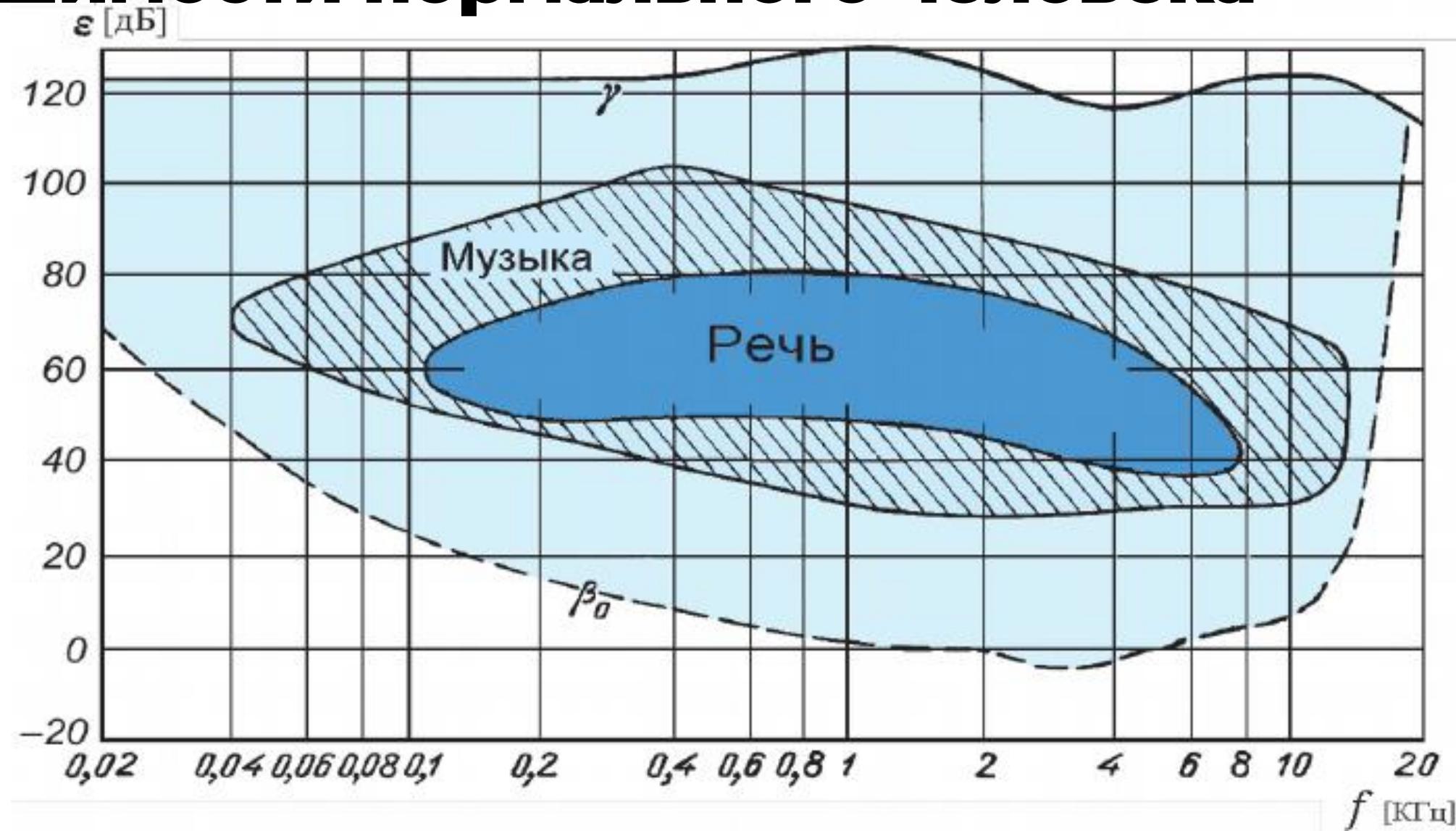
# Кодирование звуковой информации

- **Задача кодирования звуковой информации является частной задачей кодирования (восстановления) аналоговых сигналов.**
- Аналоговая информация окружает нас повсеместно и представляет собой показания, снимаемые какими-либо датчиками на протяжении какого-то времени, при этом снимаемые показания могут меняться быстро или, наоборот, медленно.
- В первом приближении, если измерять перемещения мембранны или геометрические размеры какого-либо тела (внутри микрофона), как следствие череды сжатий и разрежений воздушного пространства, то можно сделать предположение, что мы говорим о звуке. Обычно под звуком понимают такие колебания, которые оказываются слышны человеческому уху.

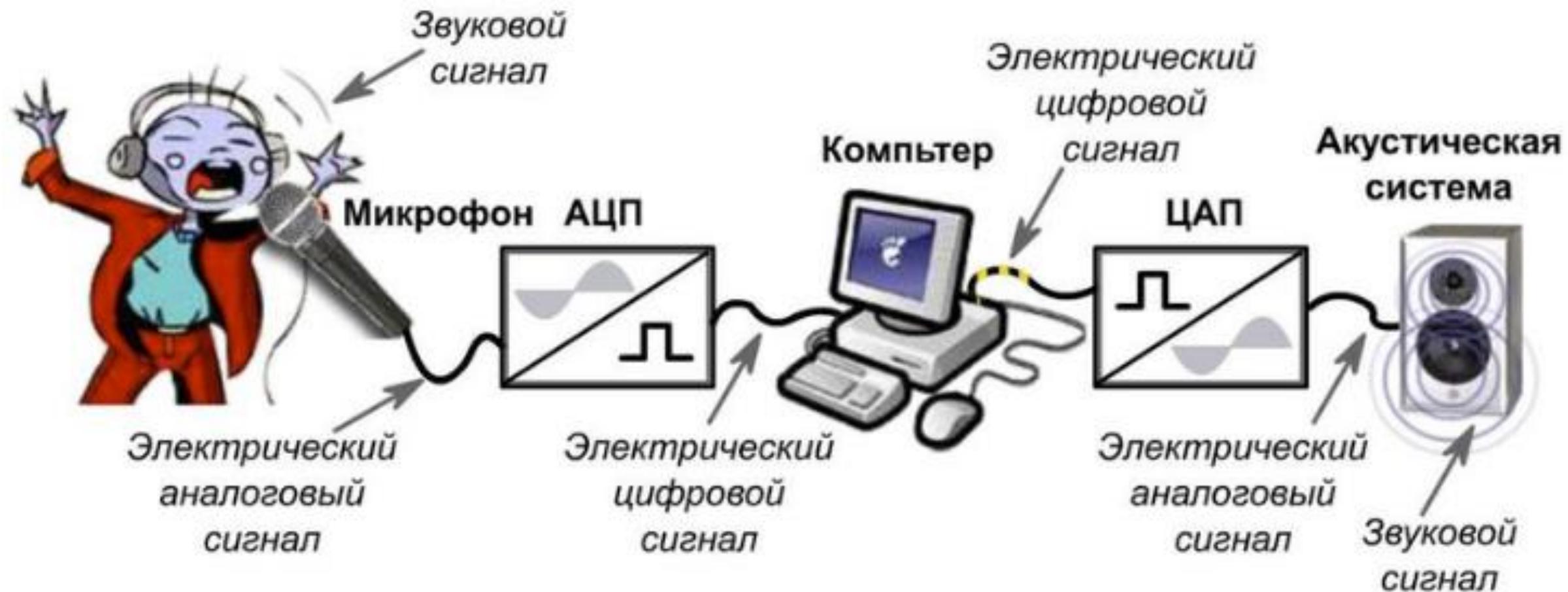
# Кодирование звуковой информации



# Спектрально-амплитудный диапазон слышимости нормального человека



# Кодирование звуковой информации



**АЦП – аналого-цифровое преобразование - оцифровка сигнала**

**ЦАП – цифро-аналоговое преобразование – преобразование в аналог**

# Кодирование звуковой информации

- В отличие от числовых, текстовых и графических данных, у звукозаписей не было столь же длительной и проверенной истории кодирования.
- В итоге **методы кодирования звуковой информации двоичным кодом далеки от стандартизации.**
- **Множество отдельных компаний разработали свои корпоративные стандарты**, но если говорить обобщенно, то можно выделить два основных **направления:**
  - Метод FM (Frequency Modulation);
  - Метод таблично-волнового (Wave-Table).

# Кодирование звуковой информации

## Метод FM (Frequency Modulation)

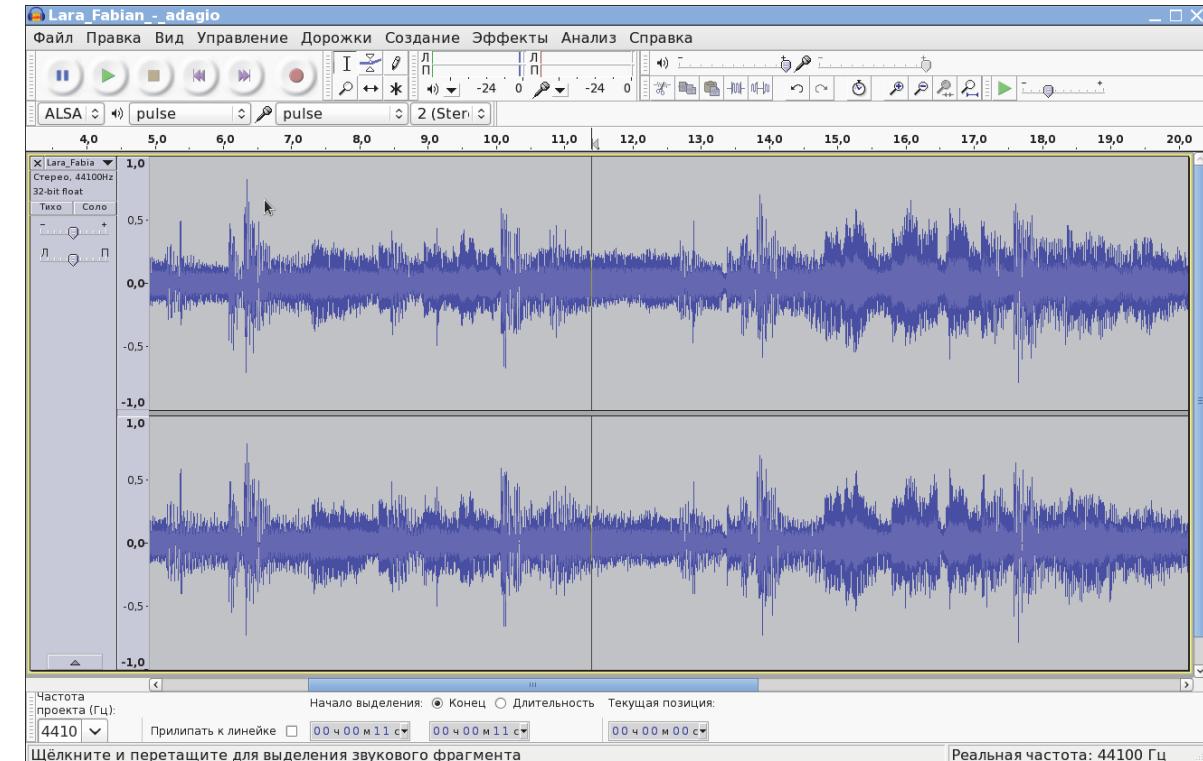
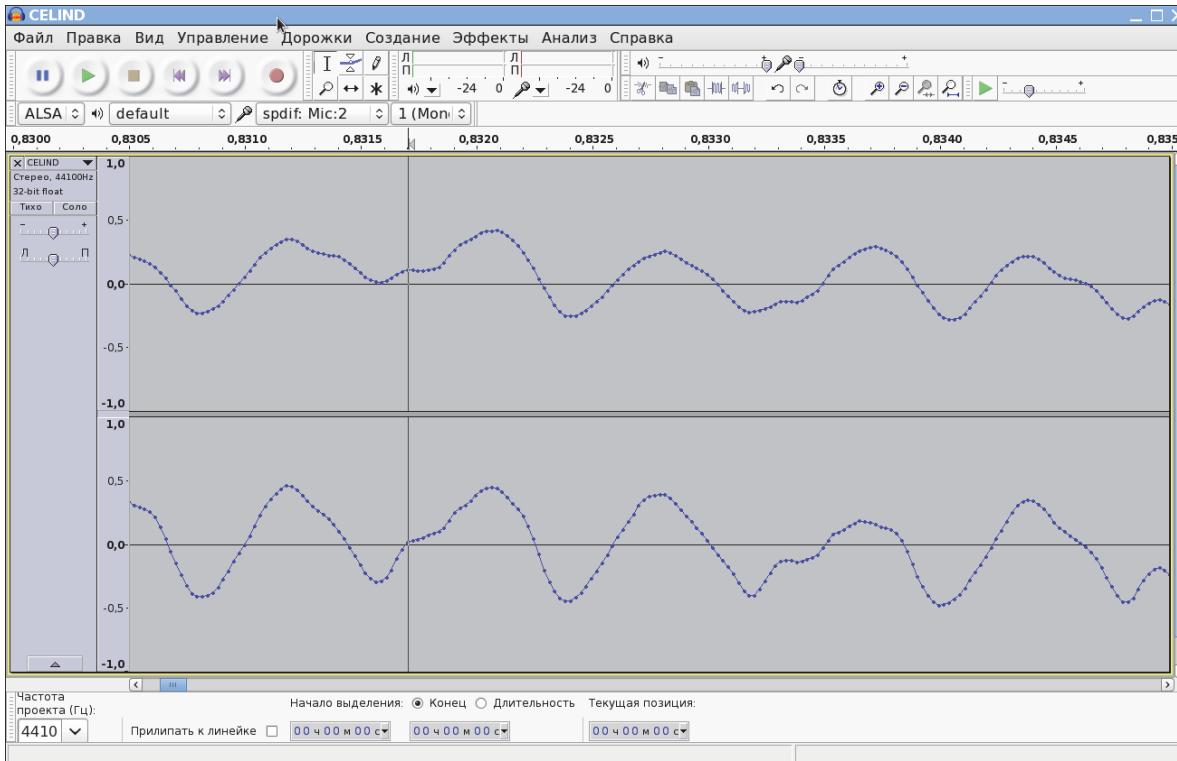
- Метод FM (Frequency Modulation) основан на том, что теоретически любой сложный звук можно разложить на последовательность простейших гармонических сигналов разных частот, каждый из которых представляет собой правильную синусоиду, а следовательно, может быть описан числовыми параметрами, то есть кодом.
- В природе звуковые сигналы имеют непрерывный спектр, то есть являются аналоговыми. Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальные устройства — аналогово-цифровые преобразователи (АЦП).
- Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют цифро-аналоговые преобразователи (ЦАП). При таких преобразованиях неизбежны потери информации, связанные с методом кодирования, поэтому качество звукозаписи обычно получается не вполне удовлетворительным и соответствует качеству звучания простейших электромузикальных инструментов с окрасом, характерным для электронной музыки.
- В то же время, данный метод кодирования обеспечивает весьма компактный код, и потому он нашел применение еще в те годы, когда ресурсы средств вычислительной техники были явно недостаточны.

# Кодирование звуковой информации

## Метод таблично-волнового (Wave-Table)

- **Метод таблично-волнового (Wave-Table) синтеза** лучше соответствует современному уровню развития техники. Если говорить упрощенно, то можно сказать, что **где-то в заранее подготовленных таблицах хранятся образцы звуков для множества различных музыкальных инструментов** (хотя не только для них). В технике такие образцы называют сэмплами. **Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры, характеризующие особенности звука.**
- Поскольку в качестве образцов используются «реальные» звуки, то качество звука, полученного в результате синтеза, получается очень высоким и приближается к качеству звучания реальных музыкальных инструментов.

# Окно аудиоредактора Audacity



# Процесс оцифровки аналогового сигнала

- Для более лёгкого понимания **процесса «оцифрования» аналогового сигнала**, а именно, что этот процесс **происходит в три простых шага**:
- **Первый шаг.** На выходе каждого измерения получается некоторое числовое (аналоговое) значение амплитуды.
- Непрерывная звуковая волна разбивается на отдельные участки по времени, для каждого устанавливается своя величина амплитуды. Каждой ступеньке присваивается свой уровень громкости звука, который можно рассматривать как набор возможных аналоговых состояний. Этот процесс называется **дискретизацией**.
- **Дискретизация** – преобразование непрерывной функции в дискретную. (Получение из непрерывной функции значений её отсчётов, взятых в отдельные моменты времени, определяемые частотой дискретизации.)

# Процесс оцифровки аналогового сигнала

- **Второй шаг.** Так как информация в ЭВМ не хранится в аналоговом виде, то значения измерений (амплитуды отсчётов) квантуются (по факту измеряются с конечной точностью, которая определяется разрядностью аналого-цифрового преобразователя, – АЦП).
- Результат этих измерений уже можно представить в виде некоторого конечного числа (которое и является значением указанного отсчёта), представимого в памяти ЭВМ в цифровом виде.
- **Квантование** (англ. quantization) – в информатике – разбиение диапазона значений непрерывной или дискретной величины (например, амплитуды уровня сигнала) на конечное число интервалов.

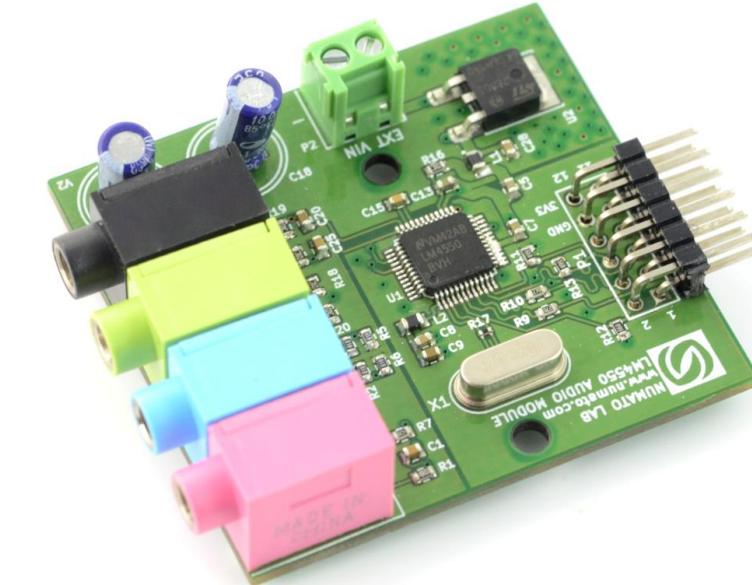
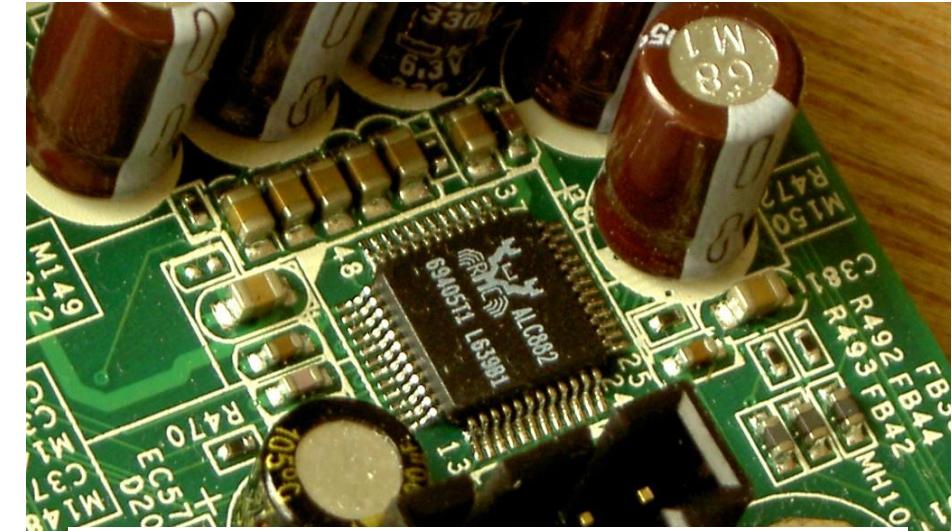
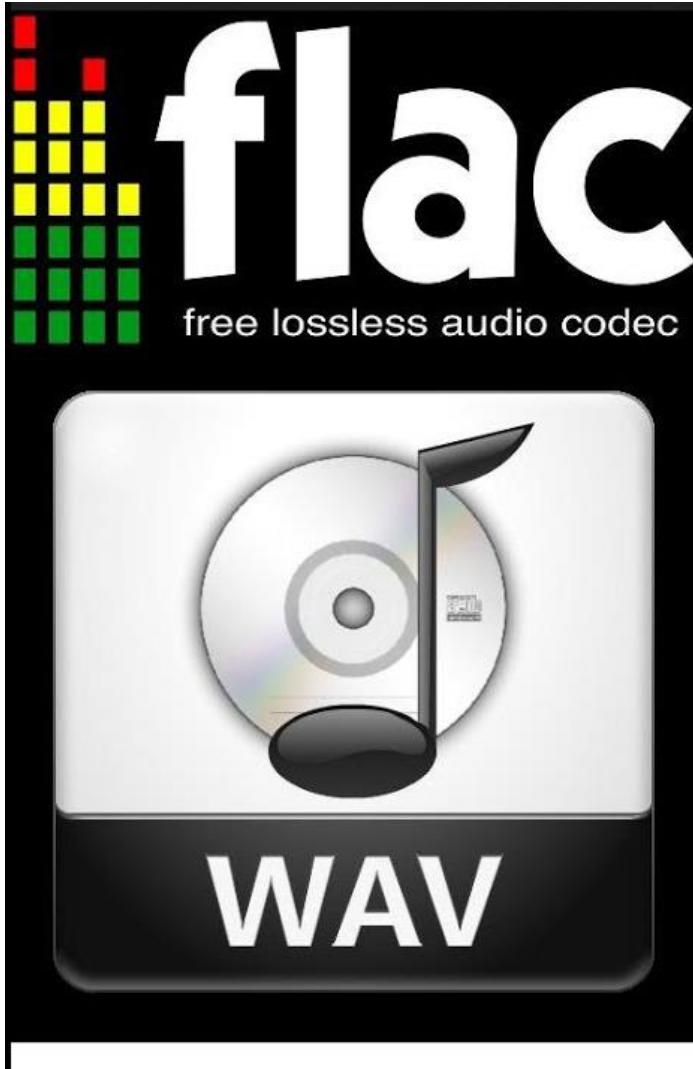
# Процесс оцифровки аналогового сигнала

- **Третий шаг** – это кодирование. В ряде случаев кодирование как таковое отсутствует и все полученные на этапе квантования значения записываются в память одно за другим по мере проведения измерений.
- Если действия для всех трёх шагов собрать вместе, то в литературе этот алгоритм называются как «Импульсно-кодовая модуляция» (ИКМ, англ. *Pulse Code Modulation*, PCM) или линейной ИКМ.

# Процесс оцифровки аналогового сигнала

- Более сложные «алгоритмы оцифрования» аналогового сигнала чаще называются алгоритмами сжатия или «кодеками», потому процесс всегда можно поделить на два шага:
  - **первый** – непосредственно получение ИКМ-последовательности и
  - **второй шаг** – сжатие полученных значений тем или иным алгоритмом (кодеком).

# Аудио кодеки



# Воспроизведение аудио



WAV/AIFF



MP3/AAC/OGG



FLAC/ALAC

# Разновидности цифровых аудиоформатов

- **Формат файла** определяет структуру и особенности представления звуковых данных при хранении на запоминающем устройстве ПК. Для устранения избыточности аудиоданных используются аудиокодеки, при помощи которых производится сжатие аудиоданных.
- **Выделяют три группы звуковых форматов файлов:**
  - аудиоформаты без сжатия, такие как WAV, AIFF
  - аудиоформаты со сжатием без потерь (APE, FLAC)
  - аудиоформаты со сжатием с потерями (MP3, Ogg)
- Особняком стоят модульные музыкальные форматы файлов, созданные синтетически или из семплов заранее записанных живых инструментов. Они в основном служат для создания современной электронной музыки (MOD). Также сюда можно отнести формат MIDI, который не является звукозаписью, но с помощью секвенсора позволяет записывать и воспроизводить музыку, используя определённый набор команд в текстовом виде.

# Разновидности цифровых аудиоформатов

- **WAV** - стандартный формат с дискретизацией 16 бит/44,1 кГц, применяемый во всех CD. Отличное качество, но размер файлов огромен.
- **Мр3** - популярный формат со сжатием с малым размером файла и далеко не самым высоким качеством звучания. Удобен для хранения музыки.
- **AAC** - формат со сжатием и потерями, как и MP3, но с лучшим звучанием. Используется для скачивания с iTunes (с битрейтом 256 кбит/с) и трансляций с YouTube.
- **Flac** - формат со сжатием без потерь с поддержкой частот дискретизации, совместимых с Hi-Res; размер файлов меньше, чем у WAV, метаданные хранятся

# Формат MP3



- MP3 (более точно, англ. MPEG-1/2/2.5 Layer 3; но не MPEG-3) — это разработанный командой MPEG формат файла для хранения аудиоинформации.
- MP3 является одним из самых распространённых и популярных форматов цифрового кодирования звуковой информации. Он широко используется в файлообменных сетях для оценочного скачивания музыкальных произведений. Формат может проигрываться практически во всех популярных операционных системах, на большинстве портативных аудиоплееров, а также поддерживается всеми современными моделями музыкальных центров и различных плееров.
- В формате MP3 используется алгоритм сжатия с потерями, разработанный для существенного уменьшения размера данных, необходимых для воспроизведения записи и обеспечения качества воспроизведения звука, близкого к оригинальному (по мнению большинства слушателей), но с ощутимыми потерями качества при прослушивании на качественной звуковой системе. Принцип сжатия заключается в снижении точности некоторых частей звукового потока, что практически неразличимо для слуха на повсеместно распространённой аппаратуре низкой точности воспроизведения звука (например, доминирующее большинство портативных устройств, звуковых карт, музыкальных центров, автомагнитол и прочей не специальной аппаратуры), а также для людей старшего возраста, в связи с естественными возрастными изменениями слухового аппарата, однако в большинстве случаев чётко различимы на аудиотехнике высокой точности воспроизведения. Данный метод называют перцепционным кодированием.



# Кодирование видеоинформации

# Кодирование видеоинформации

- **Видеоинформация включает последовательность кадров и звуковое сопровождение,** поэтому кодирование видеоинформации еще более сложная проблема, чем кодирование звуковой информации, так как нужно позаботиться не только о дискретизации непрерывных движений, но и о синхронизации изображения со звуковым сопровождением.
- В настоящее время для этого используется множество различных форматов, например один из основных форматов это AVI (Audio-Video Interleaved – чередующееся аудио и видео).

# Размер видеофайла в формате AVI

- Объем видеофайла примерно равен произведению количества информации в каждом кадре на число кадров. Число кадров вычисляется как произведение длительности видеоклипа  $\Delta t$  на скорость кадров  $v$ , то есть их количество в 1 с:

$$V = N * M * C * v * \Delta t.$$

- При разрешении  $800 \times 600$  точек, разрядности цвета  $C = 16$ , скорости кадров  $v = 25$  кадров/с, видеокlip длительностью 30 с будет иметь объем:**

$$\begin{aligned}800 * 600 * 16 * 25 * 30 &= 5\ 760\ 000\ 000 \text{ бит} = 576 * 10^7 \text{ бит} = \\&= 720\ 000\ 000 \text{ байт} = 72 * 10^7 \text{ байт} = \\&= 703\ 125 \text{ Кбайт} = 686,645 \text{ Мбайт.}\end{aligned}$$

- Это много для такого короткого видеофрагмента, поэтому на практике применяются различные способы компрессии, то есть сжатия звуковых и видеокодов.

- **Для уменьшения размера файлов AVI начали применять** способы сжатия, предложенные MPEG (Moving Pictures Experts Group – группа экспертов по движущимся изображениям). В частности, **стандарт MPEG** описывает несколько популярных в настоящее время форматов записи звука.
- Например, при записи в формате MP3 при практически том же качестве звука требуется в десять раз меньше памяти, чем при использовании формата WAV. Существуют специальные программы, которые преобразуют записи звука из формата WAV в формат MP3.
- **Наиболее популярным стал разработанный стандарт MPEG-4**, применение которого позволяло записать полнометражный цветной фильм со звуковым сопровождением на компакт-диск обычных размеров и качества.

# Видеокодек

- **Кодек — это на самом деле соединение двух слов: кодер и декодер (co / dec).** Что они делают? Проще говоря, поскольку видеофайлы очень большие, вам нужен способ сделать их меньше. Кодек кодирует, сжимая данные для хранения или отправки, а затем распаковывает для воспроизведения или редактирования.
- **Кодек** — это компьютерный код, который выполняет свою функцию всякий раз, когда файл вызывается программным обеспечением. Кодеки также можно использовать в физическом оборудовании, например в камере, для преобразования входящего видео и звука в цифровой формат.
- Это происходит в режиме реального времени, либо в точке захвата, либо в точке воспроизведения. Однако, если вы не являетесь инженером по вещанию, то вам придётся положиться на свой компьютер или устройство для выбора кодека. Аппаратное обеспечение сжимает ваши видео и аудио данные для просмотра, передачи или хранения.

# Видеоформат

- Формат файла похож на контейнер.
- Внутри контейнера находятся данные, которые были сжаты с использованием определенного кодека.  
Каждому типу контейнеров соответствует перечень кодеков, способных упаковывать и распаковывать медиаданные в нужных форматах. Так, видео в контейнере AVI может быть сжато кодеками DivX, MPEG-2, MPEG-4.
- Тип медиаконтейнера легко определить по расширению видеофайла. В обиходе тип медиаконтейнера часто называют форматом видеофайла. Известные примеры контейнеров – AVI, MKV, ASF, MP4, FLV. Данные, сжатые при помощи одного и того же кодека, могут храниться в контейнерах разных типов. К примеру, кодек MPEG-4 применяется для сжатия видеопотока в контейнерах типа MP4, MKV, MXF, OGG и QuickTime(MOV).

MPEG2

MPEG 4

H.264  
MPEG-4/AVC

H.265  
HEVC  
High Efficiency Video Coding

# Сравнение форматов записи видео на диски

Формат	Разрешение	Стандарт кодирования		Совместимость с DVD-плеером
		видео	аудио	
VCD	352×288 PAL 352×240 NTSC	MPEG-1	MPEG-1	всегда
SVCD	480×576 PAL 480×480 NTSC	MPEG-2	MPEG-1	иногда
DVD	720×576 PAL 720×480 NTSC	MPEG-2	MPEG-1, AC3	всегда
XVCD	720×576 PAL 720×480 NTSC	MPEG-1 или MPEG-2	MPEG-1	иногда
DivX	640×480	MPEG-4	MP3, WMA	иногда
HDTV 720p	1280×720	MPEG-4 H.264	MP3, WMA, AC3 или др.	BD-плеер
HDTV 1080i	1920×1080 (i – чересстрочная развёртка)	MPEG-4 H.264	MP3, WMA, AC3 или др.	BD-плеер
AVCHD 720p	1280×720 (p – прогрессивная развёртка)	MPEG-4 v.10 (AVC/H.264)	PCM (7.1) или AC3 (5.1)	нет
AVCHD 1080i	1920×1080	MPEG-4 v.10 (AVC/H.264)	PCM (7.1) или AC3 (5.1)	нет
4K UHDTV (2160p)	3840×2160	HEVC(H.265)	AAC	нет
8K UHDTV (4320p)	7680×4320	HEVC(H.265)	AAC	нет

# Форматы цифрового кодирования и сжатия

- CCIR 601 (ITU-T)
- M-JPEG (ISO)
- MPEG-1 (ISO)
- **MPEG-2** (ISO)
- **MPEG-4** (ISO)
- H.261 (ITU-T)
- H.263 (ITU-T)
- **H.264** (ITU-T и ISO)
- **H.265** или HEVC
- DV
- Ogg-Theora
- **DivX**
- **XviD**
- RealMedia
- и др.

# Наиболее популярные форматы для видео

- **AVI (Audio Video Interleave)** – старый формат, разработанный Microsoft. Поддерживается большинством популярных браузеров, работающих в системах Windows, Macintosh и Linux. Сжатие данных меньше, чем в MP4.
- **MP4 (полностью – MPEG-4 Part 14)** – формат, совместимый с большинством браузеров и поддерживаемый сайтами потокового видео, в частности, YouTube. Видео и аудиодорожки сжимаются раздельно. Видео упаковывается кодеком MPEG-4. При сжатии обеспечивается высокое качество видео (HD) при относительно небольшом размере файла.

# Наиболее популярные форматы для видео

- **MPG, MPEG, MP2, MPE, MPV** – форматы, характеризующиеся низким уровнем сжатия. Качество невысокое, а при редактировании и сохранении готового файла может ухудшиться. В этих форматах обычно записывают видео, которые впоследствии не нужно будет редактировать.
- **MOV** – формат, разработанный Apple. Видео сохраняется в хорошем качестве, но файл занимает много места. Файлы воспроизводятся с помощью проигрывателя QuickTime Player, есть версии этой программы для macOS и Windows.

# Наиболее популярные форматы для видео

- **МKV** – формат, позволяющий записать в один файл несколько звуковых дорожек и субтитров. Файлы можно смотреть как на компьютере, так и на мобильных устройствах.
- **WebM** – формат, позволяющий получать видео небольшого размера среднего качества. Видео в таком формате подходят для YouTube и других сайтов потокового видео на платформе HTML5.
- **OGG** – формат файлов потоковой передачи через интернет-каналы. Качество видео и звука лучше, чем у файлов WebM, но скорость передачи ниже.

# K-Lite Codec Pack

- **K-Lite Codec Pack** — универсальный бесплатный пакет кодеков (кодировщиков- декодировщиков) и утилит для просмотра и обработки аудио- и видеофайлов.

[https://www.codecguide.com/download\\_kl.htm](https://www.codecguide.com/download_kl.htm)

The screenshot shows the 'Codec Guide' website with a yellow header featuring a clapperboard icon and the text 'Codec Guide Your source for quality codecs'. A sidebar on the left contains links for General (News, Links, Guides), Support (Forum, F.A.Q.), K-Lite Codec Pack (Information, Download), and Other downloads (Overview, VLC Player, Windows Media Codecs, Media Foundation). The main content area is titled 'Download K-Lite Codec Pack' and describes four variants: Basic, Standard, Professional, and Ultimate. The 'Basic' variant is highlighted as 'Small but extremely powerful!' and includes a list of supported formats and features like subtitle display and hardware accelerated video decoding.

**Codec Guide**  
Your source for quality codecs

**Download K-Lite Codec Pack**

There are four different variants of the K-Lite Codec Pack. Ranging from a very small bundle that contains only the most essential decoders to a large and more comprehensive bundle. The global differences between the variants can be found below. Detailed differences can be found on the [comparison of abilities](#) and [comparison of contents](#) pages.

These codec packs are compatible with Windows Vista/7/8/8.1/10. Old versions also with XP. The packs include both 32-bit and 64-bit codecs, so they work great on both x86 and x64 variants of Windows!

**Basic**

**Small but extremely powerful!**

Already contains everything you need to play **all** common audio and video file formats.

Supports playback of:

- AVI, MKV, MP4, FLV, MPEG, MOV, TS, M2TS, WMV, RM, RMVB, OGM, WebM
- MP3, FLAC, M4A, AAC, OGG, 3GP, AMR, APE, MKA, Opus, Wavpack, Musepack
- DVD and Blu-ray (after decryption)
- and **many more** audio and video formats!

Provides lots of useful functionality, such as:

- Subtitle display
- Hardware accelerated video decoding
- Audio bitstreaming
- Video thumbnails in Explorer
- File association options
- Broken codec detection
- and much more!

**Download Basic**

[Contents](#)

[Changelog](#)



# Информатика

Тема: Кодирование информации

**Благодарю  
за внимание**

**КУТУЗОВ** Виктор Владимирович

# Список использованных источников

1. Рабочая программа по дисциплине «Информатика» / Кутузов В.В. – Могилев : Белорусско-Российский университет, 2023
2. Фотографии и картинки взяты с сайтов Яндекс.Картинки и Гугл.Картинки, иконки с flaticon.com
3. Информатика. Базовый курс. 2-е издание / Под ред. С. В. Симоновича. — СПб.: Питер, 2005. — 640 с
4. Закляков В. Ф. Информатика: учеб. для вузов – 5-е изд., перераб. и доп. – М.: ДМК Пресс, 2021. – 750 с.  
[http://learn2prog.ru/informatika/dmk/inf5\\_077.pdf](http://learn2prog.ru/informatika/dmk/inf5_077.pdf)
5. Яшин, В. Н. Информатика : учебник / В.Н. Яшин, А.Е. Колоденкова. — Москва : ИНФРА-М, 2022. — 522 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1069776. - ISBN 978-5-16-015924-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1853592>
6. Поляков К. Ю. Информатика. 8класс / К. Ю. Поляков, Е. А. Еремин.— М. : БИНОМ. Лаборатория знаний, 2017. — 256 с.
7. Волк, В. К. Информатика. Вводный курс для студентов IT-специальностей : учебное пособие / В. К. Волк. – Курган : Изд-во Курганского гос. ун-та, 2020. – 218 с.  
[http://dspace.kgsu.ru/xmlui/bitstream/handle/123456789/5674/Волк-ВК\\_2020\\_УП.pdf?sequence=1&isAllowed=y](http://dspace.kgsu.ru/xmlui/bitstream/handle/123456789/5674/Волк-ВК_2020_УП.pdf?sequence=1&isAllowed=y)
8. Петрунина Е.Б. Лекции по информатике: Учеб.-метод. пособие. – СПб.: НИУ ИТМО; ИХиБТ, 2014. – 105 с  
<https://books.ifmo.ru/file/pdf/1599.pdf>
9. Краткое объяснение кодирования текстовой информации. Информатика  
<https://bingoschool.ru/manual/kratkoe-obyasnenie-kodirovaniya-tekstovoj-informacii.-informatika/>

# Список использованных источников

10. Информатика, 10 класс. Урок № 14. Кодирование текстовой информации  
<https://resh.edu.ru/subject/lesson/5225/conspect/203083/>
11. Кодирование текстовой информации <https://www.sites.google.com/site/ivanovsinform/teoreticeskij-material/kodirovanie-informacii/kodirovanie-tekstovoj-informacii>
12. Юникод  
<https://ru.wikipedia.org/wiki/%D0%95%D0%BD%D0%86%D0%BA%D0%BE%D0%BC%D0%BE%D0%B4>
13. Плоскость (Юникод) [https://ru.wikipedia.org/wiki/Плоскость\\_\(Юникод\)#Основная\\_многоязычная\\_плоскость](https://ru.wikipedia.org/wiki/Плоскость_(Юникод)#Основная_многоязычная_плоскость)
14. Таблица символов Юникода  
<https://unicode-table.com>
15. Как закодировать рисунок двоичным кодом  
<https://pcznatok.ru/kompyutery/kak-zakodirovat-risunok-dvoichnym-kodom.html>
16. Растворное кодирование  
[https://spravochnick.ru/informatika/kodirovanie\\_informacii/rastrovoe\\_kodirovaniye/](https://spravochnick.ru/informatika/kodirovanie_informacii/rastrovoe_kodirovaniye/)
17. Как избавиться от мёртвых пикселей на ЖК-мониторе?  
<https://speedcamupdates.ru/obzory/kak-vyglyadit-bityj-piksel-na-monitore.html>
18. Конвертер цветов  
<https://csscolor.ru/>
19. Конвертор цветов  
<https://colorscheme.ru/color-converter.html>

# Список использованных источников

20. Color fonts  
<http://colorfonts.wtf>
21. Растровая графика  
[https://ru.wikipedia.org/wiki/Растровая\\_графика](https://ru.wikipedia.org/wiki/Растровая_графика)
22. Звукозапись  
<https://wiki2.org/ru/Звукозапись>
23. Аналого-цифровые и цифро-аналоговые преобразователи  
<https://slide-share.ru/nalogo-cifrovie-ifro-analogovie-preobrazovateli-316355>
24. Звуковые карты  
<https://ppt-online.org/287222>
25. Цифровые аудиоформаты  
[https://ru.wikipedia.org/wiki/Цифровые\\_аудиоформаты](https://ru.wikipedia.org/wiki/Цифровые_аудиоформаты)
26. MP3, AAC, WAV, FLAC: рассказываем обо всех форматах аудиофайлов <https://www.audiomania.ru/content/article-7314.html>
27. Формат MP3  
<https://ru.wikipedia.org/wiki/MP3>
28. Видео  
<https://ru.wikipedia.org/wiki/Видео>
29. Как выбрать лучший формат для видео  
<https://www.movavi.ru/support/how-to/what-video-format-is-the-best.html>

# Список использованных источников

30. Панюкова, Е.В. Информатика : учеб.-метод. пособие / Е.В. Панюкова, Э.В. Егорова. – Тольятти : Изд-во ТГУ, 2012. – 148 с. <https://dspace.tltsu.ru/bitstream/123456789/310/1/Егорова%201-85-11.pdf>