

Money



Money

Sergej Kurakin



**“If I had a dime for every time I've
seen someone use FLOAT to
store currency, I'd have
\$999.997634.” - IEEE 754 Joke**

Bill Karwin

<https://twitter.com/billkarwin/status/347561901460447232>

**Why
Money
matters?**



Growing Global Ecommerce!

<https://www.shopify.com/enterprise/global-ecommerce-statistics>

**New ecommerce
solutions created!**

**New payment methods
created!**

**Many computers process
money.**



**Occasionally you
will work with
money.**



M₃ O₁ N₁ E₁ Y₄

Money is any item or verifiable record that is generally accepted as payment for goods and services and repayment of debts, such as taxes, in a particular country or socio-economic context.

**The first usage of the
term came from
Mesopotamia circa 3000
BC**

**First stamped coins were
minted around 650–600
BC**

Decoupling of the US dollar from gold in 1971.

<https://en.wikipedia.org/wiki/Money>

9.99 USD

9.99 USD

9.99 USD

₪ ¢ \$ € ₪ ₮ ₱ ₲ ₩ ¥ zł

https://en.wikipedia.org/wiki/Currency_symbol

Currency pair



https://en.wikipedia.org/wiki/Currency_pair

**Quotation of the relative
value of a currency unit
against the unit of
another currency.**

EUR/USD 1.2500

or

EURUSD 1.2500

Fun fact

- GBP/USD - Cable
- EUR/CHF - Swissy
- EUR/USD - Fiber
- EUR/GBP - Chunnel
- USD/CAD - Loonie
- AUD/USD - Aussie
- USD/JPY - Gopher
- GBP/JPY - Guppy
- EUR/JPY - Yuppy
- USD/RUB - Barney

ISO 4217

https://en.wikipedia.org/wiki/ISO_4217



Alpha code: JPY

Numeric code: 392

Number of Digits: 0

Name: Japanese yen

Alpha code: AUD

Numeric code: 036

Number of Digits: 2

Name: Australian dollar

Alpha code: BHD

Numeric code: 048

Number of Digits: 3

Name: Bahraini dinar

Alpha code: UYW
Numeric code: 927
Number of Digits: 4
Name: Unidad provisional

Alpha code: XTS

Numeric code: 963

Number of Digits: ?

**Name: Code reserved for
testing**

Alpha code: XXX

Numeric code: 999

Number of Digits: ?

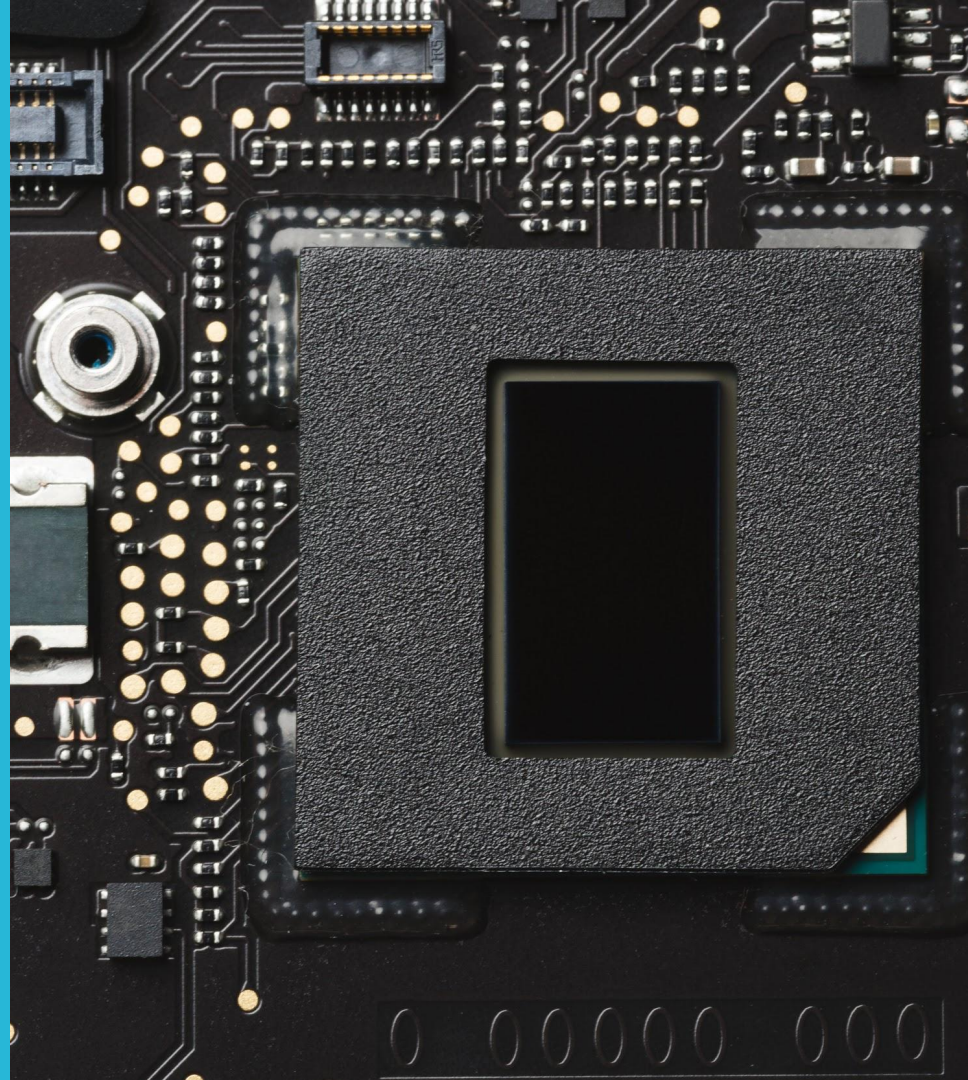
Name: No currency

Unofficial Currencies
JEP, CNH, GGP, IMP, KID,
NIS, NTD, PRB, SLS, RMB,
TVD, ZWB

**Cryptocurrencies
unofficially have Codes
and Number of Digits
6-8-12-18**



Money in Computers



What Computers do?

- Operate
- Store
- Output

Single Currency

Multiple Currencies

**Amount type: float,
integer or string**

Floats issue

0.1+0.2 ~~===~~ 0.3

Integer conversion

3.59 USD => 359 *USD*

3.59 BHD => 3590 *BHD*

String arithmetics?

“3.59” USD

“3.59” BHD

Arithmetic Operations

5 USD + 5 EUR = ? ???

5 USD != 5 EUR

5 USD / 3 =

~~1.666666666(6)~~

1.67 USD + 1.67 USD +

1.66 USD

0.05 USD 70% / 30%

~~0.035 USD / 0,015 USD~~

0.04 USD / 0.01 USD

Currency conversion.

256.12 USD => ????.?? EUR



Pattern: Value Object

Value object is a small object that represents a simple entity whose equality is not based on identity.

**There is a thing (Entity) or
a description (Value).**

Should be small.

Should be immutable.

**Value should be
Replaceable.**

```
class Age {  
    private $age;  
    public function __construct(int $age)  
    {  
        // some validation  
        $this->age = $age;  
    }  
    public function getAge(): int  
    {  
        return $this->age;  
    }  
}
```

```
final class Money {  
    private $amount;  
    private $currency; // should be a value object  
    public function __construct(int $amount,  
string $currency)  
    {  
        // some validation  
        $this->amount = $amount;  
        $this->currency = $currency;  
    }  
    ...  
}
```

```
final class Money {  
    ...  
    public function getAmount(): int  
    {  
        return $this->amount;  
    }  
    public function getCurrency(): string  
    {  
        return $this->currency;  
    }  
    ...  
}
```



```
final class Money {  
    ...  
    public function equals(Money $other): bool  
    {  
        if ($this->getCurrency() !== $other->getCurrency()) {  
            return false;  
        }  
        if ($this->getAmount() !== $other->getAmount()) {  
            return false;  
        }  
        return true;  
    }  
    ...  
}
```

```
final class Money {  
    ...  
    public function add(Money $other): Money  
    {  
        if ($this->getCurrency() !== $other->getCurrency()) {  
            throw new \Exception('Currencies not equal.');        }  
        return new Money(  
            $this->getAmount() + $other->getAmount(),  
            $this->getCurrency()  
        );  
    }  
    ...  
}
```

Currency conversion



PHP Side

Integer or String

BCMath

floats as strings

<https://www.php.net/manual/en/book.bc.php>

GMP

integers

<https://www.php.net/manual/en/ref.gmp.php>

Money PHP

<https://github.com/moneyphp/money>

Ulabox Money

<https://github.com/ulabox/money>

Do It Yourself?



Storing Money

Amount
Currency*

Fixed Decimal

Big Integer (Unsigned)

String

**Check Database
Abstraction.**



Pattern: Embedded Value

**Maps the values of an
object to fields in the
record of the object's
owner.**

<https://martinfowler.com/eaCatalog/embeddedValue.html>

```
class Product
* price: Money
```

```
Products
* price_amount
* price_currency
```



Formatting Money Output

```
money_format (  
    string $format ,  
    float $number  
) : string
```

Platform dependent.
Locale dependent.

Money PHP :: Formatting

<http://moneyphp.org/en/stable/features/formatting.html>

Currency Microformats

<http://microformats.org/wiki/currency>
<https://en.wikipedia.org/wiki/Microformat>

Front-End Libraries

**Consult with Front-End on
format**

Avoid floats

Keep consistency



Other Languages

Ruby

<https://github.com/RubyMoney/money>
integers

Go

<https://github.com/Rhymond/go-money>
integers

JavaScript

<https://github.com/sarahdayan/dinero.js>
integers

Java*

JSR 354

<https://github.com/JavaMoney>

Python

<https://github.com/carlospalol/money>

<https://github.com/vimeo/py-money>

decimal

Discussion!



Patterns of Enterprise Application Architecture
Implementing Domain-Driven Design
SQL Antipatterns: Avoiding the Pitfalls of Database Programming

Money - <https://en.wikipedia.org/wiki/Money>
Currency pair - https://en.wikipedia.org/wiki/Currency_pair

A study of 4 Money class designs

ISO 4217 https://en.wikipedia.org/wiki/ISO_4217
<https://www.currency-iso.org/en/home.html>

Thanks!

