Information:

- ist „Rohstoff“ der Informatik, hat eigenständige Dimension
  (vgl. Länge, Zeit, elektrische Ladung
  – ist jedoch nicht im SI-System enthalten)
- läßt sich nicht messen,
  sie wird berechnet
- wird durch Nachrichten übertragen
- Eine **Nachricht** ist eine Folge diskreter Zeichen oder Zustände, die zur Übertragung von Information dienen können (nicht: müssen – vgl. Photopapier)

Syntax von Programmiersprachen

Bei der Codierung von Algorithmen in Programmiersprachen werden drei Aufgabenbereiche unterschieden:

- **Die Syntax** (Form)

  Regelwerk zur Beschreibung der zulässigen Zeichenketten in einem Programm

  („Wie sage ich es meinem Rechner?“ „Was sagte der Lehrer?“)

- **Die Semantik** (Bedeutung)

  Lehre der korrekten Interpretation

  („Der Lehrer sagte der Bürgermeister ist ein Esel.“)

- **Die Pragmatik** (Wirkung)

  Lehre vom Auslösen von Handlungen und Ereignissen, von der Effektivität von Information – der schwierigste Aufgabenbereich

  („Die Tür ist hinter Ihnen.“)

Syntax von Programmiersprachen

Beispiel: Programmierung eines Weckers / einer doppelten Lichtschranke zur Erfassung nur eintretender Besucher / …

- **Syntax**

  maschinen- / compilertaugliche Codierung:

```
clock_t zeit; zeit=clock(); /*Rechenzeit*/
time_t  zeit; time(&zeit);  /*Wartezeit*/
```

- **Semantik**

  korrekt interpretierbare Codierung:

```
while(zeit1 = zeit2) vs. while(zeit1 == zeit2)
/* Wert-Setzen vs. Wert-Abfragen*/
```

- **Pragmatik**

  sinnvolle, wirksame Codierung:

```
if (zeit1 == zeit2) return; /*nie erfuellt!*/
```

Syntax von Programmiersprachen

- Die **Pragmatik** wird meist durch Erfahrung erlernt; es gibt auch einschlägige Literatur.
- Die **Semantik** wird meist als Text vermittelt; es gibt auch Ansätze zur formalisierten Darstellung.
- Die **Syntax** wird meist formal spezifiziert; zu ihrer Beschreibung dient seit den 1960ern als **Metasprache** (Notation) die **Backus-Naur-Form** (BNF):

  Sprachkonstrukte (auch: metalinguistische Variablen, Metavariablen, syntaktische Einheiten) werden definiert durch Zuweisung von Zeichen oder Zeichenketten, die i. d. gegebenen (Programmier-) Sprache zulässig sind (sog. Terminalzeichen).

  Zur Unterscheidung von den zugewiesenen Werten wird die Konstrukt-Bezeichnung in spitzen Klammern (**< >**) geschrieben.

  Wertzuweisungen werden durch das Zeichen **::=** angezeigt.

  Alternativen werden durch den senkrechten Strich (**|**) getrennt.

  – z.B.: **<Ziffer> ::= 0|1|2|3|4|5|6|7|8|9.**

Syntax von Programmiersprachen

In der BNF können Definitionen rekursiv erfolgen

z.B.: Variablen-Name in C, beginnend mit einem Buchstaben oder Unterstrich und bestehend aus Buchstaben, Unterstrichen oder Ziffern – nach Def. von **<Buchstabe>**:

```
<Name> ::= (<Buchstabe> | _) | <Name> <Buchstabe>
                            | <Name> _  | <Name> <Ziffer>
```

Die erweiterte BNF (extended BNF, EBNF)

- schließt nicht die Metavariablen (in spitzen Klammern < >), sondern die Terminalsymbole (in Anführungsstrichen " ") ein
- vereinfacht die Notation durch Verwendung unterschiedlicher Klammerarten.

Mit der EBNF werden Programmiersprachen durch eine Sammlung von Regeln, sog. **Produktionen**, spezifiziert.

Die Gesamtheit aller Produktionen einer Sprache wird als ihre **Grammatik** bezeichnet.

Syntax von Programmiersprachen

In der BNF können Definitionen rekursiv erfolgen, z.B.: Variablen-Name in C, beginnend mit einem Buchstaben oder Unterstrich und bestehend aus Buchstaben, Unterstrichen oder Ziffern – z.B. (nach Def. von **<Buchstabe>**):

```
<Name> ::= (<Buchstabe> | _) | <Name> <Buchstabe>
                            | <Name> _  | <Name> <Ziffer>
```

Die erweiterte BNF (extended BNF, EBNF)

- schließt nicht die Metavariablen (in spitzen Klammern < >), sondern die Terminalsymbole (in Anführungsstrichen " ") ein
- vereinfacht die Notation durch Verwendung unterschiedlicher Klammerarten .

Mit der EBNF werden Programmiersprachen durch eine Sammlung von Regeln, sog. **Produktionen**, spezifiziert.

Die Gesamtheit aller Produktionen einer Sprache wird als ihre **Grammatik** bezeichnet.

Syntax von Programmiersprachen

In der EBNF verwendete Zeichen und Ihre Bedeutung:

Zeichen	Bedeutung
::= (auch **=**)	ist definiert als
|	oder (alternativ)
.	Ende der Produktion
[X]	0 oder 1 Instanz von X
{X}	0 oder mehrere Instanzen von X
(X|Y)	Gruppierung: X oder Y
"XY"	Terminalzeichen XY
MetaVariable	Nichtterminalzeichen **MetaVariable**

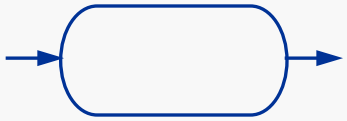
Beispiel: Variablen-Name in C (s.o.) in der EBNF:

```
Name ::= (Buchstabe|"_"){Buchstabe |"_"| Ziffer}.
```

Syntax von Programmiersprachen

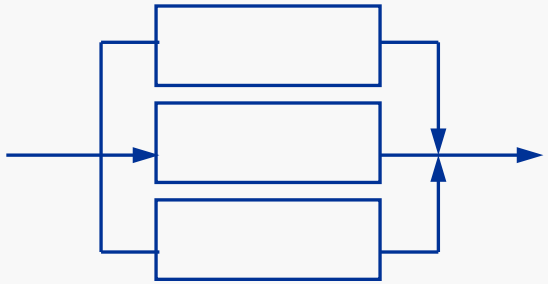
Wichtige Elemente zur grafischen Syntaxdarstellung in EBNF:

Definition



Terminalzeichen

Nichtterminalzeichen



Alternativen

Beispiel: Variablen-Name in C

```
Name ::= (Buchstabe|"_"){Buchstabe |"_"| Ziffer}.
```