



Transparenzen sicher ausgeben

■ Transparenzen und Ausgabeprobleme

Was sind „Transparenzen“ überhaupt?

In der Medienrealisation per Computer arbeiten wir objektorientiert. Ein Objekt kann zum Beispiel ein Bild, eine Grafik, ein grafisches Element oder Text sein. Traditionell ist jedes dieser Objekte *prinzipiell deckend* – seine Existenz auf einem Medium ersetzt also den Untergrund, den es verdeckt, vollkommen durch seinen eigenen Inhalt.

Die Seitenbeschreibungssprache PostScript war der Grundstein für den Siegeszug des Personal Computers in Mediengestaltung und -produktion. Auch heute noch verwenden viele Printer ebenso wie die Plattenbelichter der Druckvorstufe und der Digitaldruck diese Sprache, die eine konsistente, medienneutrale Ausgabe überhaupt erst ermöglichte.

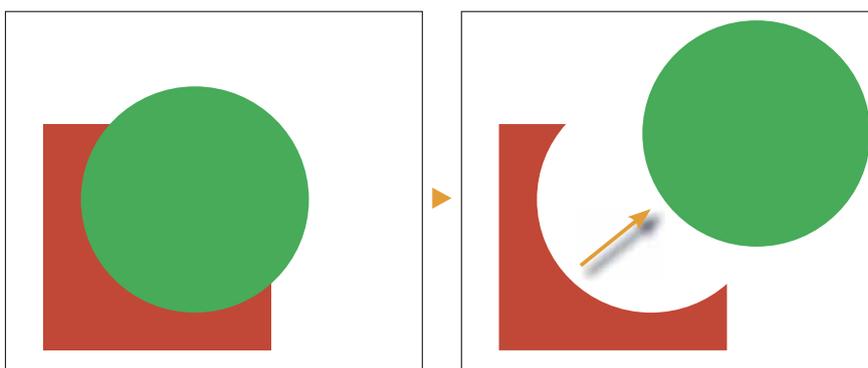
PostScript kennt aber keine Transparenzen. Das bedeutet: Jedes PostScript-Objekt verdeckt jedes unter ihm liegende. Warum? Als Seitenbeschreibungssprache war PostScript nur dazu ausgelegt, unidirektional Befehle an ein Ausgabegerät zu schicken und von diesem lediglich Rückmeldungen über Fehler zu erhalten, sonst nichts.

Warum sind Transparenzen problematisch?

Das Prinzip „Transparenz“ setzt aber voraus, dass die Informationen über den Untergrund unter einem neuen Objekt nicht verloren gehen, sondern weiterhin existieren – weil die Pixel des neuen Objekts mit den dort schon existierenden früherer Objekte „interagieren“ müssen. Genau dies kann PostScript nicht **1**.

Und das ist auch einer der Gründe, warum PDF seinen Siegeszug antrat. PDF ist nämlich keine Seitenbeschreibungssprache, sondern ein Dateiformat (das allerdings in seinen Grundzügen auf PostScript beruht). In einem Dateiformat gibt es die Beschränkung auf eine unidirektionale Kommunikation nicht. Applikationen können zu beliebigen Stellen der Datei springen, das Gesetz zur linearen Übertragung gilt hier nicht.

Seit PDF-Version 1.4 können PDF-Dateien echte Transparenzen **2** enthalten. Was aber sind „echte Transparenzen“? Das Problem der Realisation von Transparenzen betrifft immer nur die *Ausgabe*, nicht die Dateien. Um dort vorhandene Transparenzen ausgeben zu können, bediente man sich eines Tricks: „unechter Transparenzen“ **3**.



Das Photoshop-o-meter

Welche Photoshop-Version benötige ich für diesen Beitrag?



Was Sie in diesem Abschnitt lernen:

- Woher kommen die Probleme mit Transparenzen?
- Was sind „echte“ und „unechte“ Transparenzen?
- Was ist „Flattening“?
- Wie vermeide ich Probleme mit Photoshop-Transparenzen bei der Ausgabe?
- Wie verwende ich Pfade, Masken und Kanäle zum Export von Transparenzen?
- Wie erstelle ich Compositings im Layoutprogramm?

Warum kennt PostScript keine Transparenz?

Prinzip „Transparenz“: Interaktion zwischen Medienelementen

Dateiformat „PDF“

PDF-Dateien können seit PDF-Version 1.4 Transparenzen enthalten

1 Keine Transparenz: Hier wurde einfach ein farbiger Kreis über ein farbiges Rechteck gelegt. Der Kreis deckt das Rechteck teilweise ab. Unterhalb des Kreises wurde die Farbinformation des Rechtecks „vergessen“, die grüne Fläche des Kreises hat die gelbe Fläche des Rechtecks gelöscht. Würde man den Kreis verschieben, dann erschiene das Rechteck kreisförmig ausgeschnitten. Genau auf diese Weise arbeiten PostScript und PDF-Versionen bis 1.3.